

A NOVEL OPTIMIZATION FORMULATION FOR THE DIRECT COMPUTATION  
OF THE VOLTAGE COLLAPSE POINT

By

Wafa Elkhier Mohamed Elballa

Ahmed H. Eltom  
Professor of Electrical Engineering  
(Committee Chair)

Abdelrahman A. Karrar  
Professor of Electrical Engineering  
(Committee Co-Chair)

Gary L. Kobet  
Adjunct Professor of Electrical Engineering  
(Committee Member)

A NOVEL OPTIMIZATION FORMULATION FOR THE DIRECT COMPUTATION  
OF THE VOLTAGE COLLAPSE POINT

By

Wafa Elkhier Mohamed Elballa

A Thesis Submitted to the Faculty of the University of  
Tennessee at Chattanooga in Partial  
Fulfillment of the Requirements of the Degree  
of Master of Science: Engineering

The University of Tennessee at Chattanooga  
Chattanooga, Tennessee

August 2017

## ABSTRACT

The knowledge of the critical loading point of a power system is essential to assess its voltage stability condition. This point is traditionally obtained through continuation power flows which are relatively accurate but requires a considerable amount of processing time.

In this work, the search for the critical loading point of a certain power system is formulated as a constrained optimization problem which is solved using a robust scheme known as the Dog-leg Trust Region. This technique is mainly characterized by its reliability and fast convergence as it makes advantage of the merits of classical optimization techniques and gets rid of their limitations.

The proposed method was examined on several test systems namely the IEEE14, IEEE39, IEEE57, and IEEE118-bus systems. Besides successfully identifying the maximum loading point of these systems the algorithm achieved a considerable saving in processing time when compared to the continuation power flow analysis of PSAT.

## TABLE OF CONTENTS

ABSTRACT.....	iii
LIST OF TABLES.....	vi
LIST OF FIGURES.....	vii
1. INTRODUCTION.....	1
1.1 BACKGROUND.....	1
1.2 PROBLEM STATEMENT.....	1
1.3 OBJECTIVES.....	2
1.4 STUDY OUTLINE.....	2
2. LITERATURE REVIEW.....	3
2.1 INTRODUCTION.....	3
2.2 VOLTAGE INSTABILITY PROBLEM.....	4
2.3 V-P CHARACTERISTICS.....	7
2.4 THE CONTINUATION POWER FLOW.....	9
2.4.1 Mathematical Formulation.....	10
2.4.1.1 Predictor Step.....	11
2.4.1.2 Corrector Step.....	12
2.5 DIRECT METHODS.....	15
3. METHODOLOGY.....	17
3.1 PROBLEM FORMULATION.....	17
3.2 OPTIMIZATION ALGORITHMS [6].....	21
3.2.1 Line Search with the Steepest Descent Direction.....	21
3.2.2 Line Search with the Newton Direction.....	22
3.2.3 Trust Region Optimization Method.....	23
3.2.4 Dog-Leg Trust Region.....	26
3.3 APPLICATION OF THE DOG-LEG TRUST REGION SCHEME FOR MAXIMUM LOADING POINT IDENTIFICATION.....	28

3.4 BREAKDOWN OF COMPLETE OPTIMIZATION SCHEME.....	32
3.5 GENERATOR REACTIVE POWER LIMITS.....	36
4. RESULTS AND DISCUSSION .....	39
4.1 INTRODUCTION .....	39
4.2 TESTING THE SCHEME ON THE IEEE14-BUS SYSTEM .....	39
4.3 TESTING THE SCHEME ON THE IEEE39-BUS SYSTEM .....	41
4.4 TESTING THE SCHEME ON THE IEEE57-BUS SYSTEM .....	42
4.5 TESTING THE SCHEME ON THE IEEE118-BUS SYSTEM .....	44
4.6 COMPARING THE PERFORMANCE OF THE PROPOSED METHOD AGAINST PSAT .....	47
4.7 INCORPORATION OF GENERATOR REACTIVE POWER LIMITS .....	48
5. CONCLUSION.....	53
5.1 CONCLUSION.....	53
5.2 FUTURE WORK.....	54
REFERENCES .....	55
APPENDIX A.....	56
VITA.....	71

## LIST OF TABLES

4.1 Summary of the Suggested Method Performance.....	47
4.2 Comparison between the Performances of the Suggested Algorithm and PSAT.....	48
4.3 Maximum Loading Parameter When Including Generator Reactive Power Limits.....	49
4.4 Generated Reactive Power at P-V Buses and Their Maximum Limits for the IEEE 14-Bus System.....	50
4.5 Generated Reactive Power at P-V Buses and Their Maximum Limits for the IEEE 57-Bus System.....	50
4.6 Generated Reactive Power at P-V Buses and Their Maximum Limits for the IEEE 118-Bus System.....	51

## LIST OF FIGURES

2.1 Simple Radial System to illustrate Voltage Stability Problem.....	5
2.2 Receiving End Power, Voltage and Current as Function of Load Demand for the System of Figure 2.1.....	6
2.3 Nose Curve of the System of Figure 2.1.....	7
2.4 V-P Characteristics for Different Values of Power Factor.....	8
2.5 Continuation Power Flow Analysis.....	10
3.1 V-P Characteristics.....	18
3.2 Dog-leg Trajectory.....	27
3.3 Complete Optimization Scheme.....	35
4.1 Variations of the Objective Function through the Iterations for the IEEE14-Bus System.....	40
4.2 Trust Radius Dynamics for the IEEE14-Bus System.....	41
4.3 Variations of the Objective Function through the Iterations for the IEEE39-Bus System.....	42
4.4 Variations of the Objective Function through the Iterations for the IEEE57-Bus System.....	43
4.5 Trust Radius Dynamics for the IEEE57-Bus System.....	44
4.6 Variations of the Objective Function through the Iterations for the IEEE118-Bus System...	45
4.7 Trust Radius Dynamics for the IEEE118-Bus System.....	46

# CHAPTER 1

## INTRODUCTION

### **1.1 Background**

Voltage stability has become a major issue in today's highly developed power systems that are characterized by heavy loading conditions and which are expected to face increasingly stressed scenarios due to the integration of renewable generation. Consequently, voltage stability assessment tools are being developed to diagnose system conditions in order to come up with protective measures to eliminate problems associated with voltage instability. These analysis tools are generally concerned with the examination of two aspects namely, proximity to voltage instability and mechanism of voltage instability phenomenon [1]. Proximity to voltage collapse can be measured in terms of physical quantities such as loading level and reactive power reserve. On the other hand, key factors contributing to voltage instability are to be identified when investigating the mechanism of the phenomenon.

### **1.2 Problem Statement**

Determining the maximum loadability point of a power system is considered as a classical research topic which has received special attention since it serves as a principal voltage stability measurement. Conventional schemes employed to estimate maximum loading point also referred to as voltage collapse point, are either based on continuation power flow techniques or direct optimization methods. The former category, though known for its robustness, is



considered expensive in terms of processing time. On the other hand, proposed algorithms belonging to the latter category sometimes face convergence issues.

### **1.3 Objectives**

This work aims to develop a new direct method for the calculation of the maximum loading point. The search for this point is formulated as optimization problem that is then solved using the Dog-leg Trust Region scheme which is known for its robustness.

### **1.4 Study Outline**

The remainder of this work is organized as follows:

- Chapter Two: this chapter explains the voltage stability problem and reviews the schemes presented in the literature for maximum loading point calculation.
- Chapter Three: in this chapter, the proposed problem formulation is explained along with a detailed description of the optimization algorithm used in this work.
- Chapter Four: in this chapter, the simulation results when testing the developed algorithm on a variety of test systems are presented. Moreover, a comparison is made between the performance of the suggested scheme and a conventional continuation power flow.
- Chapter Five: This chapter summarizes the merits of the developed algorithm and provides recommendations for future research work.

## CHAPTER 2

### LITERATURE REVIEW

#### **2.1 Introduction**

Voltage stability is defined as the ability of a power system to maintain an acceptable voltage profile at all system buses when operating in normal conditions or after being subjected to a disturbance [1]. Voltage instability is primarily caused by failure of the system to meet reactive power demand which results in a progressive and uncontrollable decrease in voltage. This could be encountered due to a disturbance, a change in system conditions or an increase in load demand.

Voltage collapse, on the other hand, is a result of an accumulative process whereby a sequence of events accompanying voltage instability eventually lead to a low voltage profile in a significant part of the power system. It is highly influenced by system characteristics and conditions as well as the interactions between control and protective devices.

Voltage stability studies can be subdivided into two categories; small disturbance and large disturbance voltage stability [1]. The latter category investigates the system ability to control voltage following a large disturbance such as faults or loss of generation. This type of voltage stability must be studied using nonlinear time domain simulation which deploys proper system models. On the other hand, small disturbance voltage stability examines the ability of the system to control voltage following small disturbances such as a gradual increase in load. Steady

state analysis that uses a linearized version of system dynamic equations is used to study this category.

It is worth mentioning that despite the fact that voltage instability is a dynamic phenomenon, static analysis methods provide a deeper insight into the problem as it enables the investigation of wide range of system conditions and helps in identifying the key contribution factors. This is particularly true when examining system loadability margins and sensitivity relationships. Nevertheless, time domain simulations are more suitable for testing the coordination between protective and control devices and for investigations of specific voltage collapse situations.

## 2.2 Voltage Instability Problem

To further illustrate the voltage instability phenomenon, the simple radial system shown in figure 2.1 is to be considered. This system consists of a voltage source  $E_s$  that supplies a load  $Z_D$  through a line of impedance  $Z_L$ . The magnitude of the current flowing through the circuit can be expressed as shown in equation 2.1.

$$I = \frac{E_s}{\sqrt{(Z_L \cos\theta + Z_D \cos\phi)^2 + (Z_L \sin\theta + Z_D \sin\phi)^2}} \quad 2.1$$

The voltage at the receiving end can be expressed as:

$$V_r = Z_D I \quad 2.2$$

Consequently, the power supplied to the load is:

$$P_r = V_r I \cos\phi = Z_D I^2 \cos\phi \quad 2.3$$

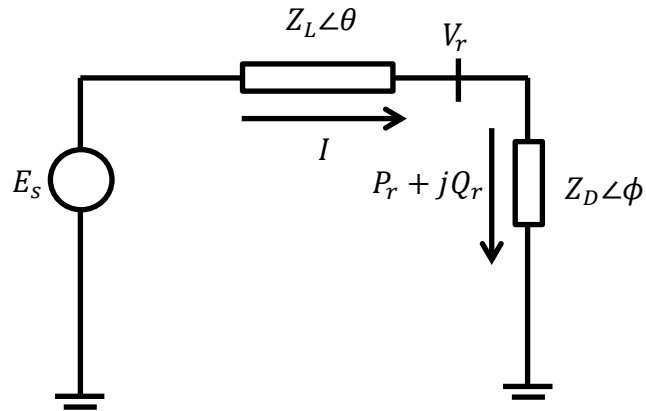


Figure 2.1 Simple Radial System to illustrate Voltage Stability Problem

Figure 2.2 depicts the normalized values of  $I$ ,  $V_r$  and  $P_r$  as a function of  $Z_L/Z_D$  for the case where  $\tan\theta = 10$  and the load power factor is 0.95. It can be noted that as the load increases, by decreasing  $Z_D$ , the current rapidly increases whereas a decrease is experienced in the value of  $V_r$ . This results in a quick increase in the delivered active power  $P_r$ . However, the power reaches a maximum when the voltage drop in the line impedance  $Z_L$  has the same magnitude as  $V_r$ . Hence, there is a maximum value for active power that can be transmitted from a constant voltage source to a load through an impedance. Beyond this maximum,  $P_r$  starts to decrease as the drop in  $V_r$  dominates over the increase in the current.

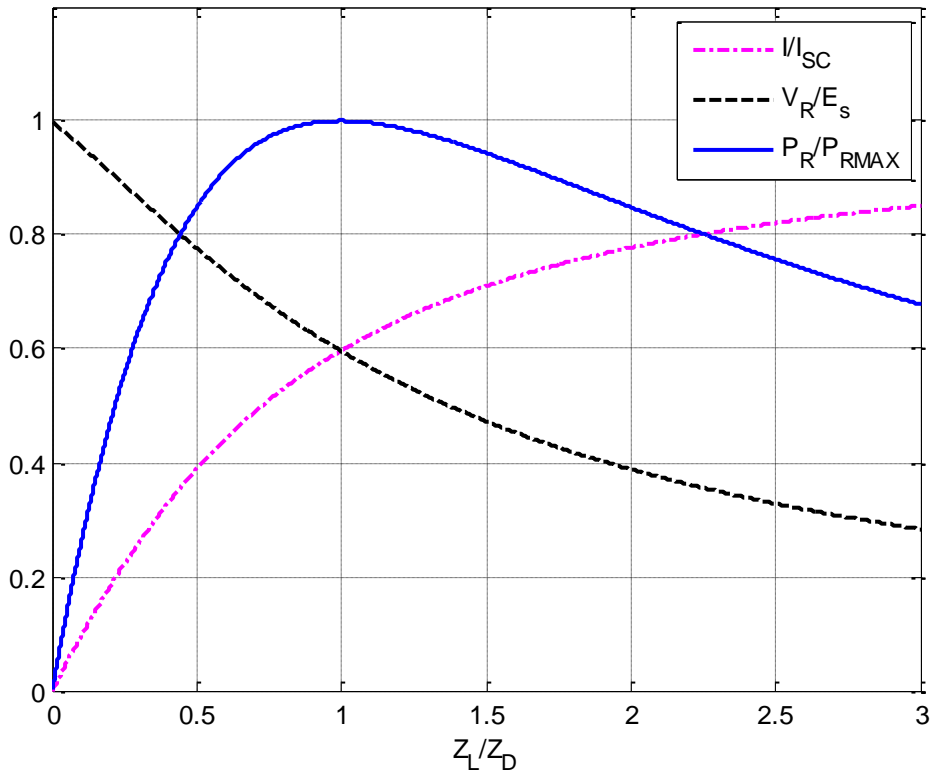


Figure 2.2 Receiving End Power, Voltage and Current as Function of Load Demand for the System of Figure 2.1

Controlling the load power past the maximum point is unstable since a decrease in load impedance would result in reducing active power. Depending on the load characteristics, the voltage might undergo a progressive decrease and the system will become unstable. For instance, with a constant power load the system experiences instability due to collapse of the load bus voltage. On the other hand, with a constant impedance load the system stabilizes at a lower voltage and power levels.

### 2.3 V-P Characteristics

When investigating voltage stability, the relationship between the delivered active power  $P_r$  and the voltage  $V_r$ , known as the nose curve, is of particular interest. For the simple power system of figure 2.1 this relationship is depicted in figure 2.3. It must be noted that only the points above the critical loading point are considered as satisfactory operating conditions while the lower part of the curve represents a region of instability.

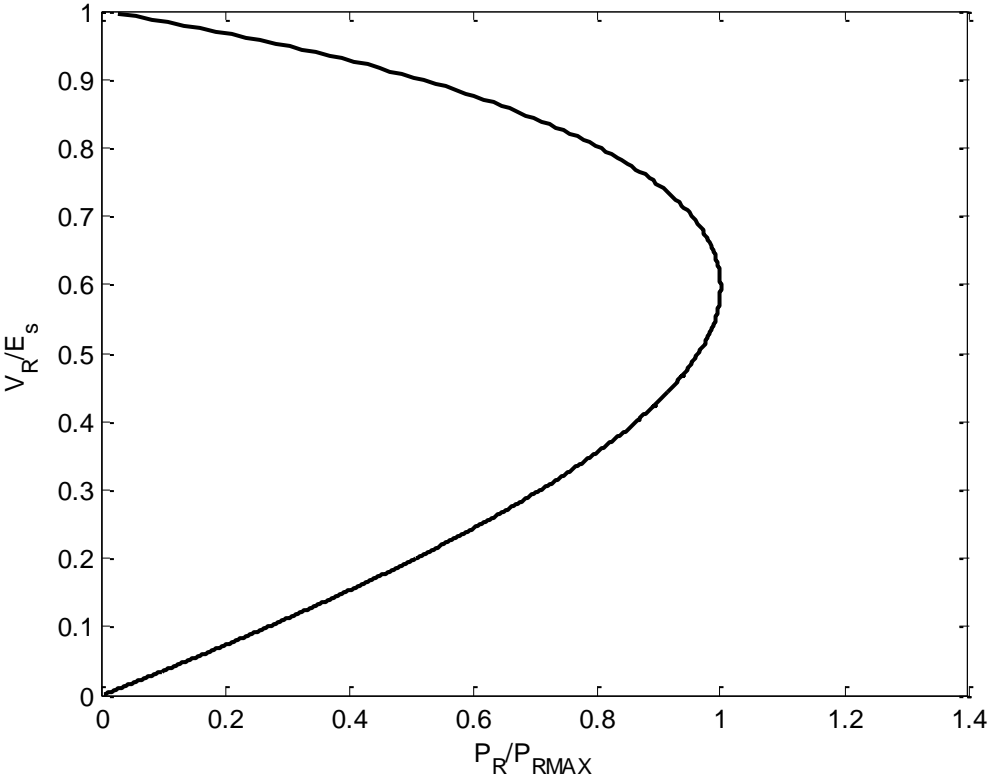


Figure 2.3 Nose Curve of the System of Figure 2.1

The V-P characteristics highly depend on the load power factor. Figure 2.4 shows these characteristics with different values of power factor. It can be noted that changing the power

factor by increasing the reactive power demand might cause the system to move from a stable to unsatisfactory or rather unstable operating condition.

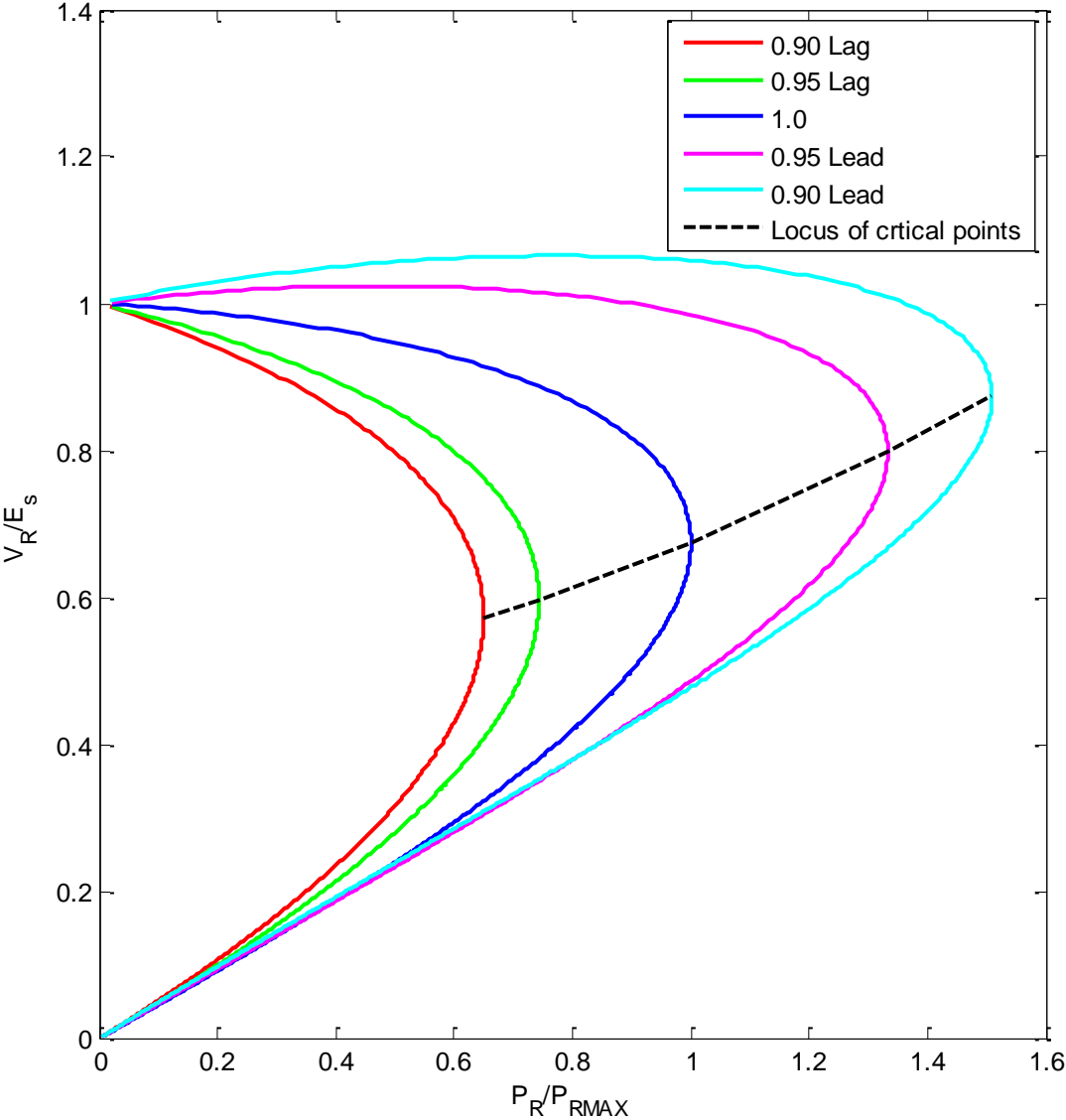


Figure 2.4 V-P Characteristics for Different Values of Power Factor

Complex power systems which have a large number of generator and load buses exhibit similar characteristics which can be determined using power flow analysis. It must be mentioned however that the Jacobian matrix of classical load flow analysis becomes singular at the point of collapse. Hence, conventional power flow schemes suffer from convergence issues when applied close to the maximum loadability point.

In literature, two types of methods have been developed to find the critical loading point of a power system. The first category is referred to as continuation approaches where a sequence of solutions of the nonlinear equations representing the power system is produced for incremental load variations. On the other hand, the second category which is termed direct methods formulates a set of nonlinear equations that characterizes the conditions of the system at the point of collapse and then tries to solve this set. The following sections provide a brief description of each category along with their associated advantages and drawbacks.

## **2.4 The Continuation Power Flow**

As mentioned earlier, the Jacobian matrix used in classic power flow analysis becomes singular at the point of collapse. To remedy this problem continuation power flow methods reformulate the load flow equations to sustain a well-condition status at all operating conditions which permits the solutions of power flow equations for the upper as well as the lower part of the V-P nose curve.

The continuation power flow (CPF) is essentially an iterative process that involves two steps known as the predictor and the corrector [1]. Referring to figure 2.5, for a specified increase in load, a tangent predictor is used to estimate the solution B from the known initial



solution A. The corrector step then arrives at solution C using conventional load flow analysis assuming the same system load as in B. The load is then increased further and the voltages are predicted using a new tangent predictor. In case the new estimated load is greater than the maximum point as in D, a corrector step with fixed load will fail to converge. In such scenario, a corrector step with fixed voltage is employed to find the solution E.

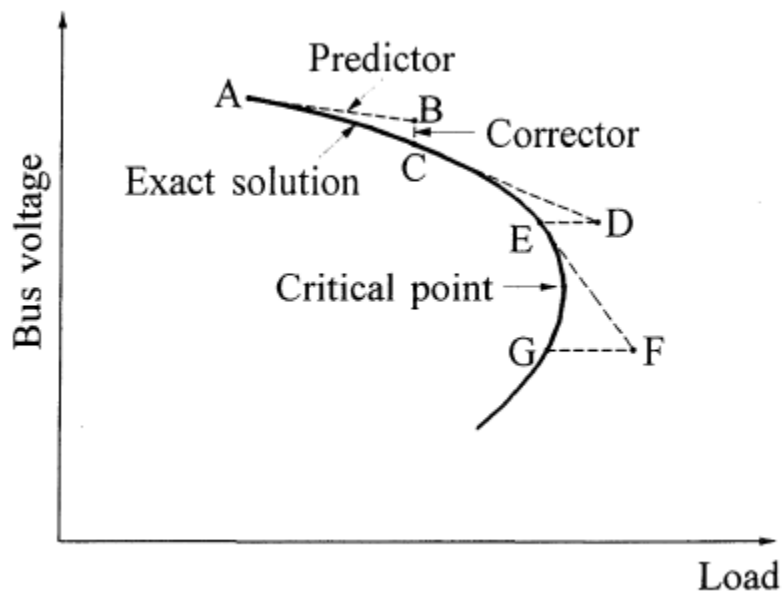


Figure 2.5 Continuation Power Flow Analysis [1]

### 2.4.1 Mathematical Formulation

Continuation power flow equations are similar to those of a conventional load flow expect for the addition of an extra parameter which represents the increase in load. This can be expressed as:

$$F(\theta, V, \lambda) = 0 \quad 2.4$$

Where:

$V$  is the vector of bus voltage magnitudes.

$\theta$  is the vector of bus voltage angles.

$\lambda$  is the loading parameter.

#### 2.4.1.1 Predictor Step

In this step, a linear approximation of equation 2.4 is employed to estimate the next solution for a change in one of the variables  $\theta$ ,  $V$  or  $\lambda$ .

The following set of linear equations is derived by taking the first derivative of equation 2.4.

$$F_{\theta}d\theta + F_VdV + F_{\lambda}d\lambda = 0 \quad 2.5$$

$$\begin{bmatrix} F_{\theta} & F_V & F_{\lambda} \end{bmatrix} \begin{bmatrix} d\theta \\ dV \\ d\lambda \end{bmatrix} = 0 \quad 2.6$$

A new equation is to be introduced to this set to make up for the addition of the loading parameter  $\lambda$ . This is done by setting one of the components of the tangent vector to -1 or +1. This component is called the continuation parameter.

$$\begin{bmatrix} F_{\theta} & F_V & F_{\lambda} \\ & e_k & \end{bmatrix} \begin{bmatrix} d\theta \\ dV \\ d\lambda \end{bmatrix} = \begin{bmatrix} 0 \\ \pm 1 \end{bmatrix} \quad 2.7$$

Where  $e_k$  is a row vector with all elements equal to zero except for that corresponding to the continuation parameter which is set to 1.

The loading parameter  $\lambda$  is initially set as the continuation parameter. However, it is changed in subsequent predictor steps to be the state variable which experiences the highest rate of change near the given solution. The sign of the slope of that parameter determines the sign of the

corresponding tangent vector. It is worth mentioning that near the loading limit the variable with greatest rate of change is typically a voltage.

When solving equation 2.7 the prediction of next solution can be calculated as:

$$\begin{bmatrix} \theta \\ V \\ \lambda \end{bmatrix} = \begin{bmatrix} \theta_o \\ V_o \\ \lambda_o \end{bmatrix} + \sigma \begin{bmatrix} d\theta \\ dV \\ d\lambda \end{bmatrix} \quad 2.8$$

The step  $\sigma$  is selected so that a load flow solution exists with the specified continuation parameter. In case a solution cannot be found in the corrector step,  $\sigma$  is reduced and the corrector step is to be repeated until reaching a solution.

#### 2.4.1.2 Corrector Step

The set of equations to be solved in this step is:

$$\begin{bmatrix} F(\theta, V, \lambda) \\ x_k - \eta \end{bmatrix} = [0] \quad 2.9$$

Where  $x_k$  represents the chosen continuation parameter and  $\eta$  is its predicted value. This set of equations can be solved using the Newton-Raphson method. It must be noted that the introduction of the new equation of  $x_k$  makes the Jacobian matrix non-singular at the loading limit which permits the continuation of the analysis beyond the loading limit point to obtain solutions corresponding the lower part of the V-P nose curve.

The sign of  $d\lambda$  indicates whether the load limit has been reached or not. This sign is positive before the collapse point and negative beyond it.

The corrector step is a vertical line in case the continuation parameter is set to be  $\lambda$ . On the other hand, if a voltage is chosen to represent this parameter, the corrector step is horizontal.

Despite its robustness and flexibility in solving power flow problems that typically face convergence difficulties, the continuation power flow is considered expensive in terms of computation time. Consequently, conventional load flow methods are sometimes used in company with continuation methods to speed up the solution process. The continuation approach is only employed near and beyond the loading limit whereas conventional load flow methods are deployed when starting from the base case.

It is worth mentioning that several modified continuation power flow methods have been proposed in the literature. These algorithms either suggested different techniques for controlling the step size or for choosing the continuation parameter around the maximum loading point. However, most of these are based on the scheme explained in this section.

Nevertheless, it must be noted that some recent publications suggested a couple of schemes that don't rely on the predictor-corrector approach. For instance, reference [2] proposed a scheme which is based on standard Newton-Raphson power flow calculations with a modified Jacobian matrix around the voltage collapse point. The scheme basically defines a critical region around the maximum loading point where the condition of the load flow Jacobian matrix exceeds a pre-specified threshold. The condition of the Jacobian matrix is defined as the ratio of its largest eigenvalue to its smallest one. A matrix is singular if at least one of its eigenvalues is zero. Consequently, when approaching the maximum loading point the Jacobian matrix becomes ill-conditioned.

Starting from the base case, the proposed scheme increases the loading at each step and solves resulting system equations using conventional Newton-Raphson method. However, the algorithm adjusts the incremental change in loading based on the condition number of the Jacobian. A large

condition number implies that the loading of the system is near its maximum and hence the step size is decreased. When entering the critical region, the algorithm perturbs the largest V-Q element in the Jacobian matrix by a small percentage  $\alpha$ . This is done to eliminate the linear dependency between the rows and columns of the matrix that causes it to be singular. Subsequently, the Newton-Raphson method can still be applied to solve the load flow problem at the maximum loading point since the Jacobian is no longer singular. The accuracy of this method however depends on the choice of the parameter  $\alpha$  since the Jacobian may still remain singular if the perturbation is too small. On the other hand, a large perturbation may lead to inaccurate solution.

Another scheme which takes advantage of a newly proposed approach for solving the load flow problem -referred to as Factored Load Flow (FLF) - is introduced in [3]. FLF is characterized by its ability to converge to complex solutions when exceeding the maximum loading point providing that a small imaginary component is added to the initial value of voltage magnitudes. The search for the maximum loading point is initialized by increasing the system load (and generation) and solving resulting system equations using FLF until the produced solution contains a non-negligible imaginary component which indicates that an infeasible point is reached. The interval between the last feasible solution and the first infeasible one is bisected using binary search strategy until its width becomes below a pre-specified tolerance. The mean value of the final interval is then defined as the maximum loading point. Despite the fact that this method is relatively simple and fast, however as in the case of Newton-Raphson approach, FLF will also breakdown if the scheme passes by the actual collapse point where the Jacobian matrix is singular.

## 2.5 Direct Methods

In [4] the maximum loadability problem is formulated as a static nonlinear optimization problem which is solved using Interior Point (IP) algorithm. The objective of the formulated optimization is to maximize the loading parameter  $\lambda$  subjected to power flow equations (equality constraints) as well as system and equipment limits (inequality constraints). In mathematical terms the problem can be represented as shown in equation 2.10.

$$\begin{aligned} \max \quad & \lambda \\ \text{s. t.} \quad & g(x, \lambda) = g(x) + \lambda D \\ & l \leq h(x) \leq u \end{aligned} \tag{2.10}$$

Where  $g(x)$  represents the standard power flow equations and  $D$  is the direction vector for load increase. The inequality constraint represents the equipment limitations such as generator reactive power limits. A Lagrangian function is then formulated using equation 2.10 and solved using Newton method.

The ease and systematic approach of incorporating and handling equality as well as inequality constraints into the problem constitutes a huge advantage of optimization methods over those which are based power flow and continuation power flow. However, the Newton method used in [4] is highly affected by the initial values of the parameters especially those of the Lagrange multipliers. The choice of these initial conditions greatly affects the convergence of the suggested algorithm.

Another direct method for the computation of the maximum loading point by formulating the calculation as a static constrained optimization is suggested in [5]. This optimization aims to

maximize the loading parameter subjected to three equality constraints which can be mathematically represented by equation 2.11

$$\begin{aligned}
 \max \quad & \rho \\
 \text{s. t.} \quad & P_g - (P_d^0 + \rho \Delta P_d) - P(e, f) = 0 \\
 & Q_g - (Q_d^0 + \rho \Delta Q_d) - Q(e, f) = 0 \\
 & V^2 - e^2 - f^2 = 0
 \end{aligned} \tag{2.11}$$

P and Q are the active and reactive power injections vectors. The subscripts g and d denote generation and load whereas  $e$  and  $f$  are the real and imaginary components of the bus complex voltage V. that is:

$$V_i = e_i + jf_i \tag{2.12}$$

Similar to the previous scheme, a Lagrangian function is formulated and solved using Newton's method.

This algorithm also turned out to be strongly influenced by the initial values of the Lagrange multipliers. As a matter of fact, the algorithm sometimes fails to converge based on the selection of these initial values. Moreover, operational inequality constraints such as generator reactive power limits are ignored when the problem is initially formulated.

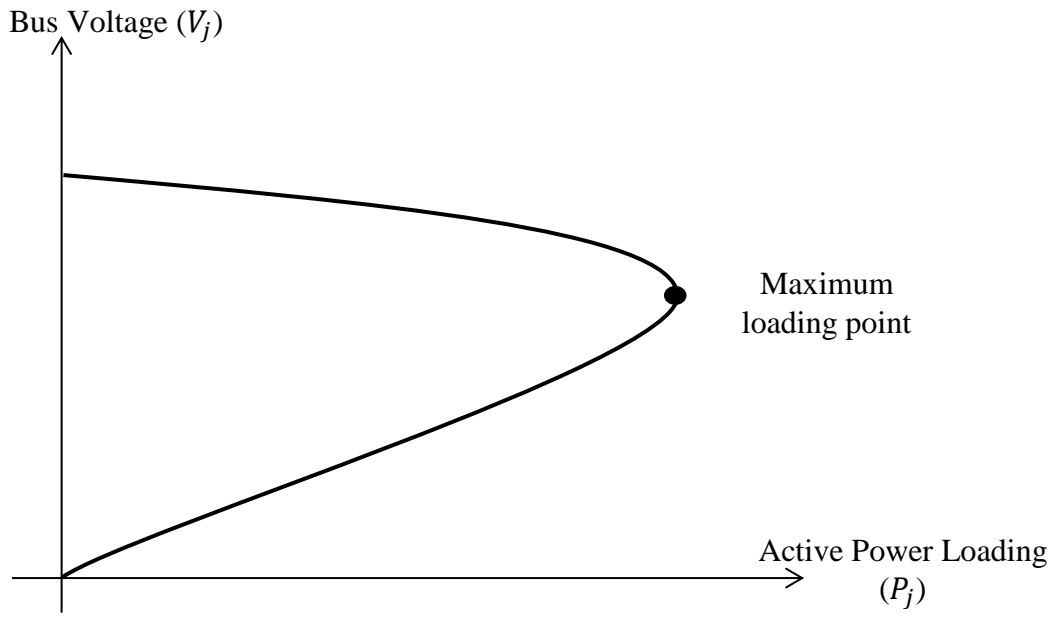
## CHAPTER 3

### METHODOLOGY

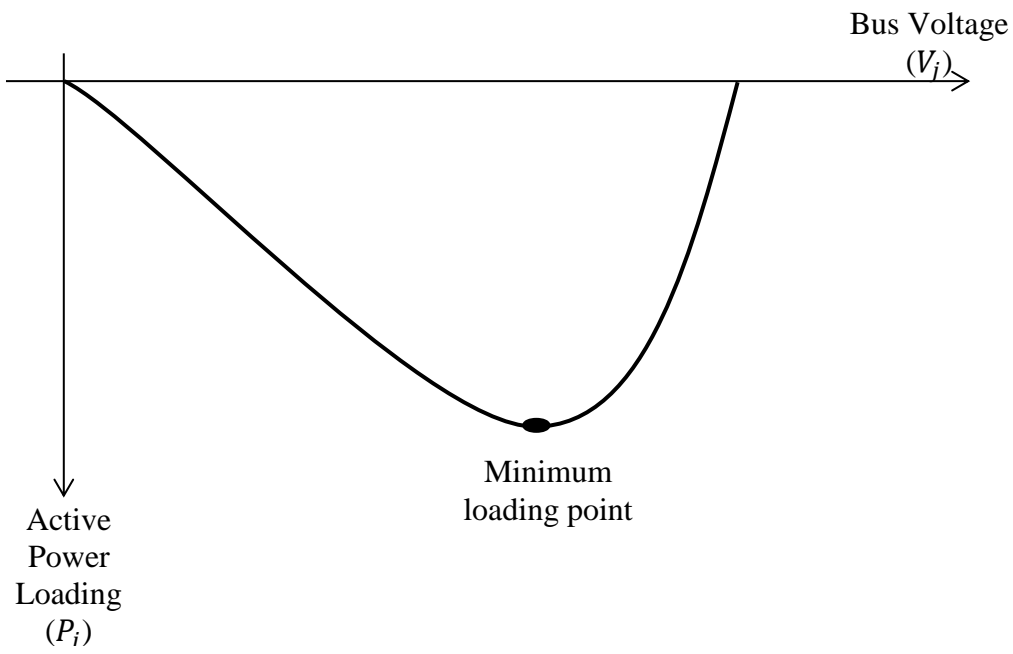
#### 3.1 Problem Formulation

The relationship between the voltage at a certain load bus and the active power injected to that bus is shown in figure 3.1 (a). As discussed in chapter (2), the maximum loading point is of particular interest since only operating points located above it are considered as satisfactory operating conditions. Redrawing the V-P characteristics while exchanging the vertical and horizontal positions of the voltage and active power results in a plot similar to that shown in figure 3.1 (b). Moreover, recognizing that the active power  $P_j$  is actually negative for loads, the maximum loading point becomes a minimum as indicated in figure 3.1 (b). The search for the critical loading point could now be regarded as an optimization problem. This optimization seeks system states (voltages and angles) at which the nose curve of figure 3.1 (b) is at its minimum. It must be mentioned however that for each load bus the V-P curve assumes a constant ratio between the reactive and active power (power factor) at all points. Moreover, the active power is assumed to increase at the same rate at all load and generation buses. These two conditions must be preserved when perusing a solution which implies that the problem is to be designed as a constrained optimization.





(a)



(b)

Figure 3.1 V-P Characteristics

In mathematical terms, assuming an n-bus system, the active power at load bus j, which is a function of system voltages and angles, is the objective function to be minimized.

$$P_j = f_j(V, \delta) \quad 3.1$$

Where:

$$V \in [V_1, V_2 \dots V_n]$$

$$\delta \in [\delta_1, \delta_2 \dots \delta_n]$$

Active and reactive power at load bus  $i$  are constrained by the power factor.

$$\frac{Q_i}{P_i} = \tan \varphi_i = \beta_i \Rightarrow \beta_i P_i - Q_i = 0 \quad 3.2$$

Likewise, active power at bus  $i$  is related to active power at bus  $j$  by:

$$\frac{P_i}{P_j} = \alpha_{ij} \Rightarrow \alpha_{ij} P_j - P_i = 0 \quad 3.3$$

A Lagrange function could now be formulated to incorporate the objective function along with these two equality constraints as shown in equation 3.4.

$$\begin{aligned} \mathcal{L}_j = P_j(V, \delta) + \sum_{i \in Load} \lambda_i (\beta_i P_i(V, \delta) - Q_i(V, \delta)) \\ + \sum_{\substack{i \in Load \\ i \in generator \\ i \neq j}} \lambda_{ij} (\alpha_{ij} P_j(V, \delta) - P_i(V, \delta)) \end{aligned} \quad 3.4$$

Where  $\lambda_i$  and  $\lambda_{ij}$  are the Lagrange multipliers.

It must be noted that in the case of a tie bus both  $\beta_i$  and  $\alpha_{ij}$  are defined to be zero in order to translate the conditions of this bus into the constraint equations. Setting  $\alpha_{ij}$  to zero implies that  $P_i$  is to be kept zero by the active power constraint (equation 3.3). Moreover, a zero  $\beta_i$  will force

$Q_i$  to be zero via the first constraint given that  $P_i$  is also zero. Hence, neither active nor reactive power injections are associated with the tie bus.

At the maximum loading point, all the partial derivatives of the Lagrange function for bus  $j$  with respect to the independent variables are equal to zero. That is:

$$\frac{\partial \mathcal{L}_j}{\partial \delta_m} = 0 \quad m \neq \text{slack bus} \quad 3.5$$

$$\frac{\partial \mathcal{L}_j}{\partial V_m} = 0 \quad m \in \text{load buses} \quad 3.6$$

$$\frac{\partial \mathcal{L}_j}{\partial \lambda_m} = 0 \quad m \in \text{load buses} \quad 3.7$$

$$\frac{\partial \mathcal{L}_j}{\partial \lambda_{mj}} = 0 \quad m \neq \text{slack bus and } m \neq j \quad 3.8$$

Equations 3.5 through 3.8 are nonlinear algebraic equations that are to be solved in order to find the system state variables  $(V, \delta)$  as well as the Lagrange multipliers  $(\lambda_m, \lambda_{mj})$  which drive  $\mathcal{L}_j$  to be minimum. Attempting to solve these equations numerically using the classical Newton-Raphson approach might cause the solution to diverge since the maximum loading point is usually far from base case conditions. This is different from the load flow problem in which the solution lies closer to the flat start point. To further clarify this argument, the following section provides a recapitulation of the most commonly used optimization algorithms along with the merits and drawbacks associated with each method. This upcoming section will also help establishing a proper context to introduce the algorithm that was chosen in this work to solve the optimization problem under study.

## 3.2 Optimization Algorithms [6]

In mathematics, optimization algorithms are used to evaluate the minimum or the maximum value of a certain objective function  $f(x)$ . The variables of this function may or may not be subjected to constraints. Generally speaking, optimization algorithms begin with an initial guess of the function variables and then generate a sequence of iterations which, when convergence is achievable, ends with a solution. In each iteration, certain information about the objective function are collected and used to formulate a stop criterion which when met ends the iterative process.

What distinguishes between different optimization algorithms is the strategy in which the algorithm moves from one iteration to another. The following subsections present some of the most frequently used strategies to evaluate the next iteration from the previous one.

### 3.2.1 Line Search with the Steepest Descent Direction

All line search methods specify a direction and a step length in each iteration. Assuming the direction to be  $p_k$  and the step length to be  $\sigma_k$ , the new iteration is defined as:

$$x_{k+1} = x_k + \sigma_k p_k \quad 3.9$$

This algorithm in particular selects  $p_k$  to be the steepest descent direction which can be determined using equation 3.10.

$$p_k = -\frac{\nabla f(x_k)}{\|\nabla f(x_k)\|} \quad 3.10$$

Where  $\nabla f(x_k)$  is the gradient of the objective function and  $\|\nabla f(x_k)\|$  is the Euclidian norm of this gradient.

Choosing the aforementioned direction guarantees a monotone decrease in the objective function. On the other hand, when selecting the step length, the ideal choice would be the global minimizer of the function:

$$\Phi(\sigma_k) = f(x_k + \sigma_k p_k) \quad 3.11$$

Nevertheless, evaluating this value may introduce some computational difficulties. Therefore, most practical line search schemes estimate an inexact minimizer of 3.11 which reduces the function  $\Phi(\sigma_k)$  to a certain degree but without a guarantee that this estimate is the actual global minimizer of the function in the pre-specified direction. For instance, a sequence of candidate step lengths may be tested. The algorithm then selects the candidate which introduces the largest decrease in  $\Phi(\sigma_k)$ .

### 3.2.2 Line Search with the Newton Direction

In this technique, the real objective function  $f(x)$  is approximated around the current iterate using the second-order Taylor expansion. The approximated function is referred to as the model function and can be expressed as shown in equation 3.12.

$$f(x_k + p_k) \approx m_k(p_k) = f(x_k) + p_k^T \nabla f(x_k) + \frac{1}{2} p_k^T \nabla^2 f(x_k) \cdot p_k \quad 3.12$$

Where  $p_k^T$  is the transpose of  $p_k$  and  $\nabla^2 f(x_k)$  is the second derivative (Hessian) of the objective function  $f(x_k)$ .

Newton direction is then found by obtaining  $p_k$  which minimizes the model function  $m_k(p_k)$ . This is done by simply finding the first derivative of the model function with respect to  $p_k$  and set it to zero which yields:

$$p_k = -(\nabla^2 f(x_k))^{-1} \nabla f(x_k) \quad 3.13$$

This will result in the exact minimizer of the model function and no additional computational effort is needed to specify the step length. Accordingly, it can be understood that the step length in this method is always set to one. It must be noted however that the reliability of the Newton direction depends on the similarity between the model function and the true objective function. In cases where the second order approximation poorly simulates the true function, this method might face troubles to converge to the actual minimizer of the objective function. On the other hand, if the two functions are very similar, the Newton method can arrive at a solution after a few iterations. This is contrary to the Steepest Descent approach where the real objective function experiences a monotone decrease from one iteration to the other, however the number of iterations needed to arrive at a solution is larger when compared the Newton method. A more robust technique that makes advantage of the merits of both methods and overcomes their limitations is introduced in the following subsection.

### 3.2.3 Trust Region Optimization Method

Similar to the Newton line search, the Trust Region method simplifies the optimization problem by using a quadratic model function in place of the real objective function. However, this technique first defines a neighborhood around the current iterate which is referred to as the trust region. The algorithm then searches for the minimizer of the model function inside that region. In mathematical terms, each iteration  $k$  in this method involves the solution of the sub-problem given by equation 3.14.

$$\min m_k(\Delta x) = f(x_k) + \Delta x^T \nabla f(x_k) + \frac{1}{2} \Delta x^T \nabla^2 f(x_k) \Delta x \quad 3.14$$

$$\|\Delta x\| \leq \Delta_k$$

Where  $\Delta_k$  is the trust region radius at step  $k$ .

Insisting on confining the step length and direction within a specific region in the vicinity of the current point helps when dealing with scenarios in which the quadratic model function is different from the actual function to be optimized. It can be noted however that the choice of the trust region radius constitutes an important factor in the performance of the algorithm. A large radius might lead to the same issues encountered when deploying the classical Newton approach as the minimizer of the model function is likely to be far from the minimizer of the actual function. On the other hand, within a small region the algorithm might proceed very slowly to the solution.

Most practical schemes start with an initial trust radius and update it based on the performance of the algorithm. More specifically, these algorithms calculate the ratio shown in equation 3.15.

$$\rho_k = \frac{f(x_k) - f(x_{k+1})}{m(x_k) - m(x_{k+1})} \quad 3.15$$

The numerator represents the actual reduction in the objective function when moving from  $x_k$  to  $x_{k+1}$  whereas the denominator is the predicted reduction based on the improvement in the model function.

A ratio close to one indicates the presence of a strong agreement between the model function and the true objective function. The scheme is then encouraged to enlarge the trust region to speed up the solution process. On the other hand, a negative ratio or a ratio near zero signifies that the model function fails to behave like the objective function within the pre-specified trust region.

As a consequence, the algorithm might consider rejecting the current step and shrinking the trust boundaries.

The performance of the trust region scheme is not only determined by the strategy in which the algorithm controls the trust radius, but also by the efficiency of the algorithm in solving the sub-problem shown in equation 3.14. A common way to solve this problem uses the so-called Cauchy point which is defined as the minimizer of the model function along the steepest descent direction. This minimizer is however restricted to be within the boundary of the trust region. The task of determining the Cauchy point is relatively easy since the model function is less complicated than the real objective function.

The unrestricted minimizer of the model function in the steepest descent direction at iteration  $k$  can be defined to be at step of  $-\sigma_k^c \frac{\nabla f(x_k)}{\|\nabla f(x_k)\|}$  where  $\sigma_k^c$  is the step length. Substituting this in 3.14 will result in:

$$m_k(\sigma_k^c) = f(x_k) - \sigma_k^c \frac{\nabla f(x_k)^T}{\|\nabla f(x_k)\|} \nabla f(x_k) + \frac{(\sigma_k^c)^2}{2} \frac{\nabla f(x_k)^T}{\|\nabla f(x_k)\|} \nabla^2 f(x_k) \frac{\nabla f(x_k)}{\|\nabla f(x_k)\|} \quad 3.16$$

The value of  $\sigma_k$  along the steepest descent direction that minimizes the model function can then be found by differentiating 3.16 with respect to  $\sigma_k$  and setting this derivative to zero. This yields:

$$\sigma_k^c = \frac{\nabla f(x_k)^T \nabla f(x_k)}{\nabla f(x_k)^T \nabla^2 f(x_k) \nabla f(x_k)} = \frac{\|\nabla f(x_k)\|^2}{\nabla f(x_k)^T \nabla^2 f(x_k) \nabla f(x_k)} \quad 3.17$$



If the value of  $\sigma_k$  turned out to be greater than the trust region radius  $\Delta_k$ , the algorithm takes the point at the boundary of the trust region which lies in the steepest descent direction. Hence the Cauchy step at the  $k^{th}$  iteration can be calculated as indicated by equation 3.18.

$$\Delta x_k^{cp} = \begin{cases} -\sigma_k^c \frac{\nabla f(x_k)}{\|\nabla f(x_k)\|} & \text{if } \sigma_k^c < \Delta_k \\ -\Delta_k \frac{\nabla f(x_k)}{\|\nabla f(x_k)\|} & \text{if } \sigma_k^c \geq \Delta_k \end{cases} \quad 3.18$$

It is worth mentioning that although solving the sub-problem defined in 3.14 using the Cauchy point guarantees convergence, the algorithm loses the chance of making advantage of the fast convergence of the Newton line search. As a matter of fact, using the Cauchy point is a slight enhancement of the Steepest Descent direction since the model function is used instead of the real objective function. However, fast convergence is only achieved when using Newton direction.

### 3.2.4 Dog-Leg Trust Region

As explained earlier, a robust scheme would incorporate the reliability of the Steepest Descent step together with the fast quadratic convergence of the Newton step. This is accomplished in the Dog-leg Trust Region algorithm which compromises between the two steps based on the trust radius.

Let's assume that  $x_k$  is the current iterate,  $x_k^{CP}$  is the Cauchy point calculated using the update of equation 3.18 and  $x_k^{NP}$  is the newton point obtained from the update in equation 3.13. The dog-leg trajectory is then defined by the two line segments joining  $x_k$ ,  $x_k^{CP}$  and  $x_k^{NP}$  respectively. This is depicted in figure 3.2. This scheme first tests if the Newton point lays inside the trust region boundaries as in the situation of the region bounded by the solid line in figure

3.2. If this turns out to be the case, then the Newton step is considered as the solution for the sub-problem 3.14. Otherwise, the point of intersection between the dog-leg trajectory and the boundaries of the trust region is taken as the next step (dotted boundary case).

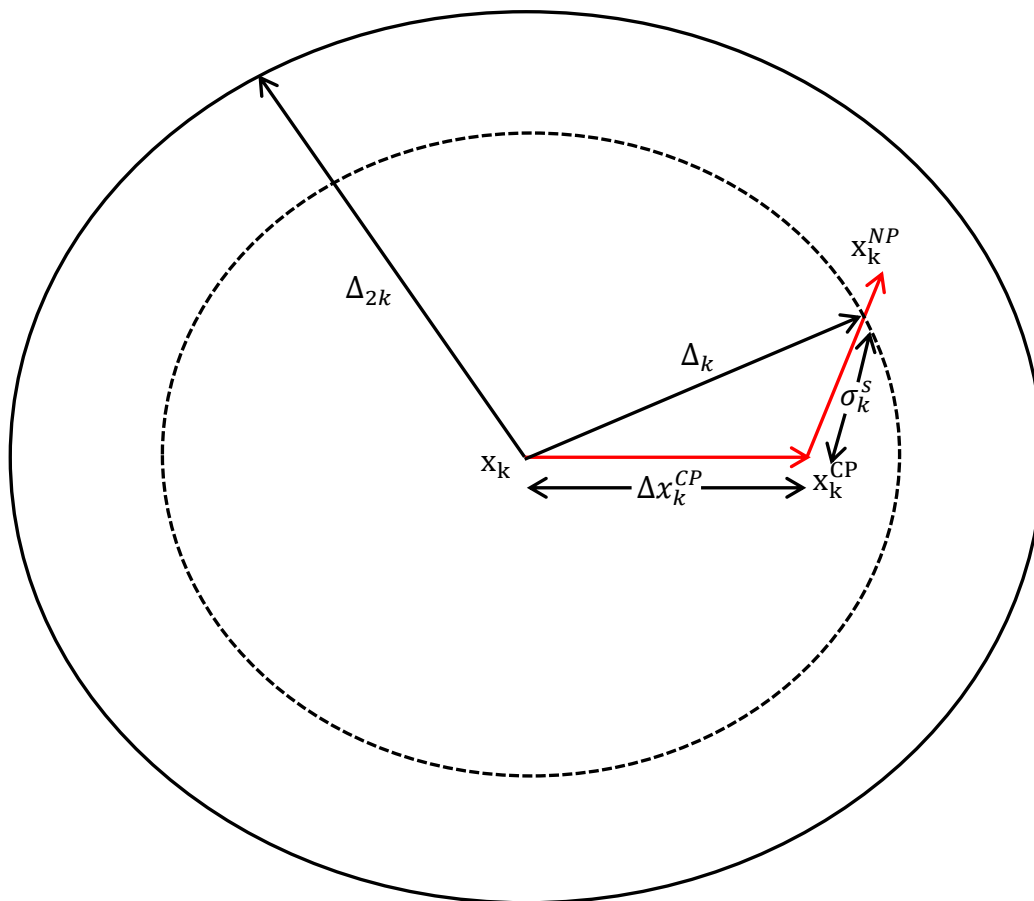


Figure 3.2 Dog-leg Trajectory

This point can be calculated by solving equation 3.19.

$$\Delta_k = \left\| \left( \Delta x_k^{CP} \right) + \frac{\sigma_k^S (x_k^{NP} - x_k^{CP})}{\|x_k^{NP} - x_k^{CP}\|} \right\| \quad 3.19$$

Where  $\frac{(x_k^{NP} - x_k^{CP})}{\|x_k^{NP} - x_k^{CP}\|}$  defines the direction along the second line segment of the Dog-leg trajectory

and  $\sigma_k^S$  is distance from  $x_k^{NP}$  to the point of intersection. The value of  $\sigma_k^S$  that solves 3.19 is:

$$\sigma_k^S = \frac{\|x_k^{CP}\|^2 - (x_k^{CP})^T x_k^{NP} + \sqrt{((x_k^{CP})^T x_k^{NP} - \|x_k^{CP}\|^2)^2 + \|x_k^{NP} - x_k^{CP}\|^2 (\Delta_k^2 - \|x_k^{CP}\|^2)}}{\|x_k^{NP} - x_k^{CP}\|} \quad 3.20$$

It is clear that the intersection point will be closer to the Cauchy point in case of narrow trust boundaries whereas large trust regions bring this point closer to the Newton step.

### 3.3 Application of the Dog-Leg Trust Region Scheme for Maximum Loading Point

#### Identification

After establishing a suitable background in optimization theory, the discussion can now proceed to handle the problem of determining the maximum loading point of a power system. The robustness of the Dog-leg Trust Region scheme is what evidently makes it the algorithm of choice in this work. This section explains how the problem under study is conditioned to be solved using the dog-leg algorithm.

The first derivatives of the Lagrange function expressed in equations 3.5 through 3.8 can be collected in a single vector which represents the gradient of the Lagrange function.

$$\nabla f(x_k) = \begin{bmatrix} \frac{\partial \mathcal{L}_j}{\partial \delta_m} \\ \frac{\partial \mathcal{L}_j}{\partial V_m} \\ \frac{\partial \mathcal{L}_j}{\partial \lambda_m} \\ \frac{\partial \mathcal{L}_j}{\partial \lambda_{mj}} \end{bmatrix} \quad 3.21$$

The length of this vector can be calculated as follows:

$$(n - 1) + n_{load} + n_{load} + (n - 2) = 2(n + n_{load}) - 3 \quad 3.22$$

Where  $n_{load}$  represents the number of load buses.

The objective function can be redefined to be:

$$F(x_k) = \frac{1}{2} \nabla f(x_k)^T \nabla f(x_k) \quad 3.23$$

It can be noted that when the Lagrange function of equation 3.4 is at its minimum,  $F(x_k)$  is zero. Hence, the objective of the optimization can be fulfilled by driving  $F(x_k)$  to zero.

The model function can now be obtained using the second order Taylor series as shown in equation 3.24.

$$\begin{aligned} m(x_k) &= \frac{1}{2} \nabla f(x_k)^T \nabla f(x_k) + \Delta x^T \nabla^2 f(x_k)^T \nabla f(x_k) \\ &\quad + \frac{1}{2} \Delta x^T \nabla^2 f(x_k)^T \nabla^2 f(x_k) \Delta x \\ &= \frac{1}{2} \nabla f(x_k)^T \nabla f(x_k) + \Delta x^T H(x_k)^T \nabla f(x_k) + \frac{1}{2} \Delta x^T H(x_k)^T H(x_k) \Delta x \end{aligned} \quad 3.24$$

The second derivative  $\nabla^2 f(x_k)$  also known as the Hessian matrix has the form shown in equation 3.25.

$$\nabla^2 f(x_k) = H(x_k) = \begin{bmatrix} \frac{\partial}{\partial \delta_s} \left( \frac{\partial \mathcal{L}_j}{\partial \delta_m} \right) & \frac{\partial}{\partial V_s} \left( \frac{\partial \mathcal{L}_j}{\partial \delta_m} \right) & \frac{\partial}{\partial \lambda_s} \left( \frac{\partial \mathcal{L}_j}{\partial \delta_m} \right) & \frac{\partial}{\partial \lambda_{sj}} \left( \frac{\partial \mathcal{L}_j}{\partial \delta_m} \right) \\ \frac{\partial}{\partial \delta_s} \left( \frac{\partial \mathcal{L}_j}{\partial V_m} \right) & \frac{\partial}{\partial V_s} \left( \frac{\partial \mathcal{L}_j}{\partial V_m} \right) & \frac{\partial}{\partial \lambda_s} \left( \frac{\partial \mathcal{L}_j}{\partial V_m} \right) & \frac{\partial}{\partial \lambda_{sj}} \left( \frac{\partial \mathcal{L}_j}{\partial V_m} \right) \\ \frac{\partial}{\partial \delta_s} \left( \frac{\partial \mathcal{L}_j}{\partial \lambda_m} \right) & \frac{\partial}{\partial V_s} \left( \frac{\partial \mathcal{L}_j}{\partial \lambda_m} \right) & \frac{\partial}{\partial \lambda_s} \left( \frac{\partial \mathcal{L}_j}{\partial \lambda_m} \right) & \frac{\partial}{\partial \lambda_{sj}} \left( \frac{\partial \mathcal{L}_j}{\partial \lambda_m} \right) \\ \frac{\partial}{\partial \delta_s} \left( \frac{\partial \mathcal{L}_j}{\partial \lambda_{mj}} \right) & \frac{\partial}{\partial V_s} \left( \frac{\partial \mathcal{L}_j}{\partial \lambda_{mj}} \right) & \frac{\partial}{\partial \lambda_s} \left( \frac{\partial \mathcal{L}_j}{\partial \lambda_{mj}} \right) & \frac{\partial}{\partial \lambda_{sj}} \left( \frac{\partial \mathcal{L}_j}{\partial \lambda_{mj}} \right) \end{bmatrix} \quad 3.25$$

The Hessian matrix has dimensions of  $(2(n + n_{load}) - 3) \times (2(n + n_{load}) - 3)$ . The reader may refer to appendix A which states the equations needed to calculate the elements of the gradient as well as the Hessian matrices.

The lower-right quadrant of the Hessian matrix turns out to be zero which can be easily seen by referring to equation 3.4. Hence,  $H(x_k)$  can be rewritten as:

$$H(x_k) = \begin{bmatrix} \frac{\partial}{\partial \delta_s} \left( \frac{\partial \mathcal{L}_j}{\partial \delta_m} \right) & \frac{\partial}{\partial V_s} \left( \frac{\partial \mathcal{L}_j}{\partial \delta_m} \right) & \frac{\partial}{\partial \lambda_s} \left( \frac{\partial \mathcal{L}_j}{\partial \delta_m} \right) & \frac{\partial}{\partial \lambda_{sj}} \left( \frac{\partial \mathcal{L}_j}{\partial \delta_m} \right) \\ \frac{\partial}{\partial \delta_s} \left( \frac{\partial \mathcal{L}_j}{\partial V_m} \right) & \frac{\partial}{\partial V_s} \left( \frac{\partial \mathcal{L}_j}{\partial V_m} \right) & \frac{\partial}{\partial \lambda_s} \left( \frac{\partial \mathcal{L}_j}{\partial V_m} \right) & \frac{\partial}{\partial \lambda_{sj}} \left( \frac{\partial \mathcal{L}_j}{\partial V_m} \right) \\ \frac{\partial}{\partial \delta_s} \left( \frac{\partial \mathcal{L}_j}{\partial \lambda_m} \right) & \frac{\partial}{\partial V_s} \left( \frac{\partial \mathcal{L}_j}{\partial \lambda_m} \right) & 0 \cdots 0 & 0 \cdots 0 \\ \frac{\partial}{\partial \delta_s} \left( \frac{\partial \mathcal{L}_j}{\partial \lambda_{mj}} \right) & \frac{\partial}{\partial V_s} \left( \frac{\partial \mathcal{L}_j}{\partial \lambda_{mj}} \right) & \vdots \quad \ddots \quad \vdots & \vdots \quad \ddots \quad \vdots \\ \frac{\partial}{\partial \delta_s} \left( \frac{\partial \mathcal{L}_j}{\partial \lambda_m} \right) & \frac{\partial}{\partial V_s} \left( \frac{\partial \mathcal{L}_j}{\partial \lambda_m} \right) & 0 \cdots 0 & 0 \cdots 0 \\ \frac{\partial}{\partial \delta_s} \left( \frac{\partial \mathcal{L}_j}{\partial \lambda_{mj}} \right) & \frac{\partial}{\partial V_s} \left( \frac{\partial \mathcal{L}_j}{\partial \lambda_{mj}} \right) & 0 \cdots 0 & 0 \cdots 0 \\ \frac{\partial}{\partial \delta_s} \left( \frac{\partial \mathcal{L}_j}{\partial \lambda_{mj}} \right) & \frac{\partial}{\partial V_s} \left( \frac{\partial \mathcal{L}_j}{\partial \lambda_{mj}} \right) & \vdots \quad \ddots \quad \vdots & \vdots \quad \ddots \quad \vdots \\ \frac{\partial}{\partial \delta_s} \left( \frac{\partial \mathcal{L}_j}{\partial \lambda_{mj}} \right) & \frac{\partial}{\partial V_s} \left( \frac{\partial \mathcal{L}_j}{\partial \lambda_{mj}} \right) & 0 \cdots 0 & 0 \cdots 0 \end{bmatrix} \quad 3.26$$

The steepest descent direction of the objective function of equation 3.23 can be found as:

$$-\nabla F(x_k) = -\nabla \left( \frac{1}{2} \nabla f(x_k)^T \nabla f(x_k) \right) = H(x_k)^T \nabla f(x_k) \quad 3.27$$

Hence, following the same reasoning explained in section (3.2.3), the Cauchy step of the problem under study can be calculated as follows:

$$\sigma_k^c = \frac{\|H(x_k)^T \nabla f(x_k)\|^2}{[H(x_k)^T \nabla f(x_k)]^T H(x_k)^T H(x_k) [H(x_k)^T \nabla f(x_k)]} \quad 3.28$$

$$\Delta x_k^{cp} = \begin{cases} -\sigma_k^c \frac{H(x_k)^T \nabla f(x_k)}{\|H(x_k)^T \nabla f(x_k)\|} & \text{if } \sigma_k^c < \Delta_k \\ -\Delta_k \frac{H(x_k)^T \nabla f(x_k)}{\|H(x_k)^T \nabla f(x_k)\|} & \text{if } \sigma_k^c \geq \Delta_k \end{cases} \quad 3.29$$

The Newton point can be obtained in the same manner explained in section (3.2.2), by differentiating equation (3.24) with respect to  $\Delta x$  and setting the derivative to zero which results in:

$$\Delta x_k^{NP} = -(H(x_k))^{-1} \nabla f_k \quad 3.30$$

The fact that the lower-right quadrant of the Hessian matrix is always zero can be exploited to avoid the inversion of the entire Hessian matrix. Equation (3.30) can be rewritten as follows:

$$\begin{bmatrix} \nabla f_k^{(1)} \\ \nabla f_k^{(2)} \end{bmatrix} = \begin{bmatrix} H_1 & H_2 & & \\ & 0 & \dots & 0 \\ H_3 & \vdots & \ddots & \vdots \\ & 0 & \dots & 0 \end{bmatrix} \begin{bmatrix} \Delta \delta^{NP} \\ \Delta V^{NP} \\ \Delta \lambda_i^{NP} \\ \Delta \lambda_{ij}^{NP} \end{bmatrix} \quad 3.30$$

Hence:

$$\nabla f_k^{(1)} = [H_1] \begin{bmatrix} \Delta \delta^{NP} \\ \Delta V^{NP} \end{bmatrix} + [H_2] \begin{bmatrix} \Delta \lambda_i^{NP} \\ \Delta \lambda_{ij}^{NP} \end{bmatrix} \quad 3.31$$

$$\nabla f_k^{(2)} = H_3 \begin{bmatrix} \Delta \delta^{NP} \\ \Delta V^{NP} \end{bmatrix} \quad 3.32$$

From equation (3.31):

$$\begin{bmatrix} \Delta \delta^{NP} \\ \Delta V^{NP} \end{bmatrix} = [H_1]^{-1} \nabla f_k^{(1)} - [H_1]^{-1} [H_2] \begin{bmatrix} \Delta \lambda_i^{NP} \\ \Delta \lambda_{ij}^{NP} \end{bmatrix} \quad 3.33$$

Substituting this in equation (3.32) yields:

$$H_3[H_1]^{-1}\nabla f_k^{(1)} - \nabla f_k^{(2)} = H_3[H_1]^{-1}[H_2] \begin{bmatrix} \Delta\lambda_i^{NP} \\ \Delta\lambda_{ij}^{NP} \end{bmatrix} \quad 3.34$$

Equation (3.34) can be used to solve for  $\begin{bmatrix} \Delta\lambda_i^{NP} \\ \Delta\lambda_{ij}^{NP} \end{bmatrix}$  which could be in turn substituted in equation (3.33) to find  $\begin{bmatrix} \Delta\delta^{NP} \\ \Delta V^{NP} \end{bmatrix}$ . The computational effort involved in this process is approximately one third less than the effort needed to find the inverse of the whole Hessian matrix.

### 3.4 Breakdown of Complete Optimization Scheme

After identifying the load bus  $j$  for which the critical loading point  $\lambda_{\max}$  is to be calculated, the scheme starts by evaluating the constants  $\beta_i$  for all load buses and  $\alpha_{ij}$  for all system buses. The vector  $x_k$  is then initialized by utilizing the load flow results of current system condition as initial values for system states  $(V, \delta)$ . Furthermore, a value of one is used as an initial trust radius whereas the maximum allowable radius is set to 10. These chosen limits sound very reasonable since  $x_k$  consists of system states and Lagrange multipliers which are not expected to take steps larger than 10.

Following the initialization process, the scheme proceeds to calculate the gradient  $\nabla f(x_k)$  and the Hessian  $H(x_k)$ . These two matrices are in turn used to evaluate the Cauchy step  $\Delta x_k^{\text{cp}}$  according to equation 3.29. In case the Cauchy point was found to lay at the boundaries of the trust region, the algorithm will take it as the step size  $\Delta x_k$  to modify the current step  $x_k$ . Otherwise, the algorithm will consider the evaluation of the Newton step  $\Delta x_k^{\text{NP}}$  as indicated by equation 3.30. As in the case of the Cauchy step, if the Newton point turns out to be inside the boundaries of the trust neighborhood then it will be considered as a valid solution. However, in

the opposite case the algorithm will find a middle ground by evaluating the point of intersection between the Dog-leg path and the trust boundaries which is defined by:

$$x_k^S = x_k^{CP} + \frac{\sigma_k^S (x_k^{NP} - x_k^{CP})}{\|x_k^{NP} - x_k^{CP}\|} \quad 3.35$$

Where  $\sigma_k^S$  is evaluated using equation 3.20.

Once the step size  $\Delta x_k$  is determined, the algorithm examines whether the new point  $x_{k+1}$  will introduce a satisfactory reduction in the true objective function. As explained earlier, this is done by calculating the ratio  $\rho_k$  using equation 3.15 where the predicted reduction in the model function of equation 3.24 can be evaluated as follows:

$$m(x_k) - m(x_{k+1}) = -(\nabla f(x_k))^T H(x_k) \Delta x + \frac{1}{2} \Delta x^T H(x_k)^T H(x_k) \Delta x \quad 3.36$$

In cases where  $\rho_k$  falls below 0.25, the algorithm rejects the current step  $\Delta x_k$ , shrinks the trust region by modifying the trust radius to be 0.25 of the current radius and attempt to evaluate a new Cauchy point based on this updated radius. On the other hand, if  $\rho_k$  exceeds 0.75, the algorithm sets the trust radius of the upcoming iteration to be 2.5 times the current radius. A reduction ratio between 0.25 and 0.75 signifies that the model function is trusted to mimic the objective function in the current trust boundaries and there is no need to modify the radius.

At the end of each iteration the algorithm checks if the value of the objective function falls below a specified tolerance in which case the iterations will be terminated and the system states at the final iteration  $x_k^f$  will be considered as the optimum solution. The maximum loading point is then calculated as:



$$\lambda_{max} = \frac{P_j^f}{P_j^o} = \frac{\sum_{k=1}^n |Y_{jk} V_j^f V_k^f| \cos(\theta_{jk} + \delta_k^f - \delta_j^f)}{P_j^o} \quad 3.37$$

Where  $P_j^o$  is the load at bus  $j$  at the base load case.

The flow chart of the scheme is depicted in figure 3.3.

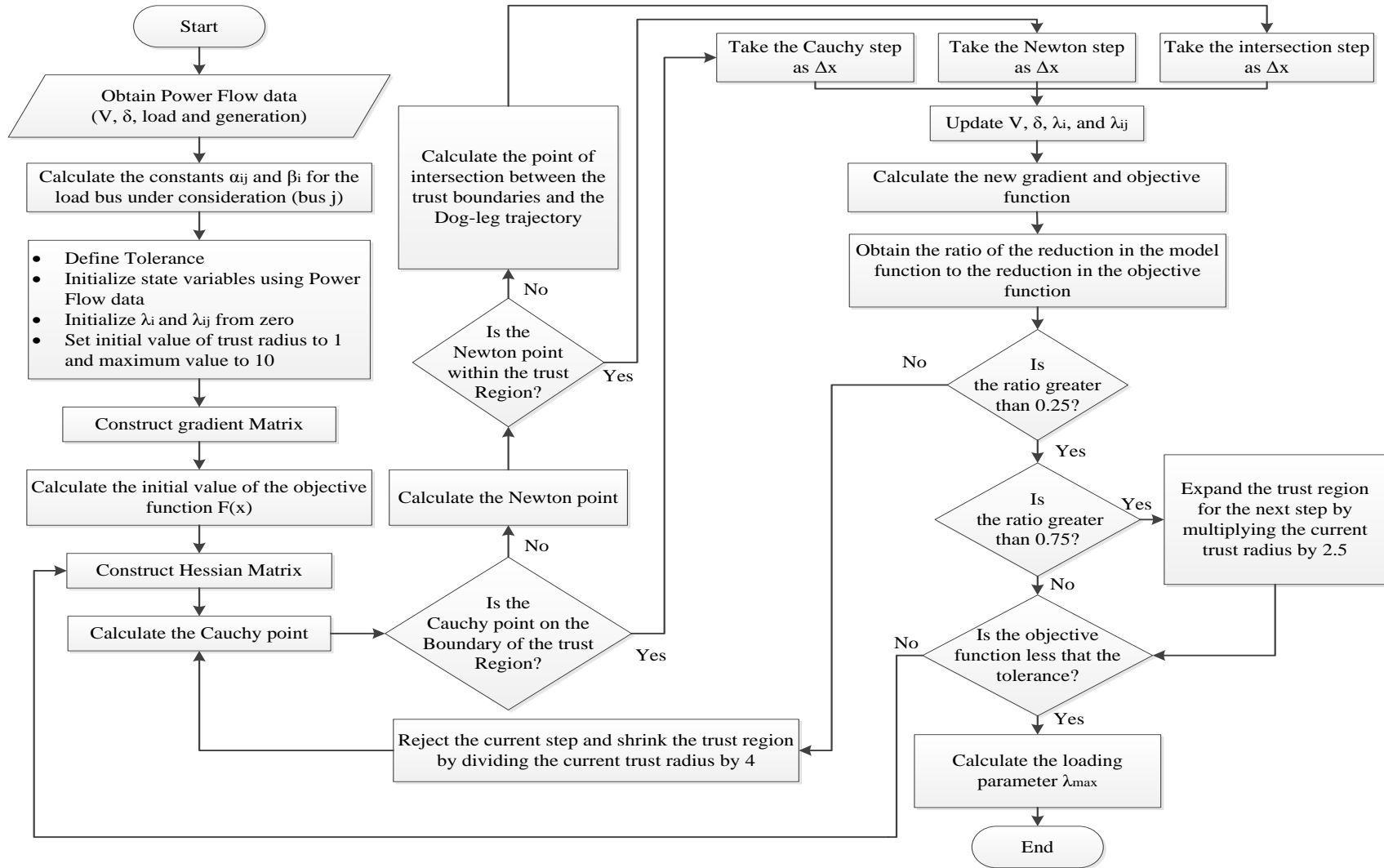


Figure 3.3 Complete Optimization Scheme

### 3.5 Generator Reactive Power Limits

The proposed algorithm could be easily modified to incorporate maximum reactive power limits of generation units. This is essentially accomplished by introducing a new constraint to the Lagrange function originally defined in equation 3.4 whenever a generator exceeds its limit. This constraint can be represented using the following equality:

$$Q_i = Q_{limit-i} - Q_{Li} \quad 3.38$$

Where:

$Q_{limit-i} \equiv$  The maximum reactive power limit at generator bus  $i$ .

$Q_{Li} \equiv$  Reactive power load connected to generator bus  $i$ .

If  $\beta_i$  is defined for generator bus  $i$  as  $\frac{Q_{Li}}{P_i}$  then using equation 3.3, the reactive power limit constraint can be re-expressed as:

$$Q_i = Q_{limit-i} - \alpha_{ij}\beta_i P_j \quad 3.39$$

When adding this term to equation 3.4, the resulting Lagrange function will be:

$$\begin{aligned} \mathcal{L}_j = & P_j(V, \delta) + \sum_{i \in Load} \lambda_i (\beta_i P_i(V, \delta) - Q_i(V, \delta)) \\ & + \sum_{\substack{i \in Load \\ i \in generator \\ i \neq j}} \lambda_{ij} (\alpha_{ij} P_j(V, \delta) - P_i(V, \delta)) \\ & + \sum_{\substack{i \in generator \text{ which} \\ \text{exceed limit}}} \lambda_{Li} (Q_{limit-i} - \alpha_{ij}\beta_i P_j - Q_i) \end{aligned} \quad 3.40$$

Where  $\lambda_{Li}$  is the Lagrange multiplier associated with generator reactive power limit constraint.

It's evident that the presence of  $\lambda_{Li}$  necessitates the expansion of the dimensions of both the gradient  $\nabla f(x_k)$  and the Hessian  $H(x_k)$  matrices to accommodate this new variable. Moreover, the voltage of the generator bus that exceeds the limit also comes into play since it is no longer constant. As a consequence, whenever a generator hits its reactive power limit the length of the gradient vector  $\nabla f(x_k)$  will increase by two while two rows and two columns are to be added to the Hessian  $H(x_k)$ . This is shown in equations 3.41 and 3.42.

$$f(x_k) = \begin{bmatrix} \frac{\partial \mathcal{L}_j}{\partial \delta_i} \\ \frac{\partial \mathcal{L}_j}{\partial V_i} \\ \frac{\partial \mathcal{L}_j}{\partial \lambda_i} \\ \frac{\partial \mathcal{L}_j}{\partial \lambda_{ij}} \\ \frac{\partial \mathcal{L}_j}{\partial \lambda_{Li}} \end{bmatrix} \quad 3.41$$

$$H(x_k) = \begin{bmatrix} \frac{\partial}{\partial \delta_s} \left( \frac{\partial \mathcal{L}_j}{\partial \delta_m} \right) & \frac{\partial}{\partial V_s} \left( \frac{\partial \mathcal{L}_j}{\partial \delta_m} \right) & \frac{\partial}{\partial \lambda_s} \left( \frac{\partial \mathcal{L}_j}{\partial \delta_m} \right) & \frac{\partial}{\partial \lambda_{sj}} \left( \frac{\partial \mathcal{L}_j}{\partial \delta_m} \right) & \frac{\partial}{\partial \lambda_{Ls}} \left( \frac{\partial \mathcal{L}_j}{\partial \delta_m} \right) \\ \frac{\partial}{\partial \delta_s} \left( \frac{\partial \mathcal{L}_j}{\partial V_m} \right) & \frac{\partial}{\partial V_s} \left( \frac{\partial \mathcal{L}_j}{\partial V_m} \right) & \frac{\partial}{\partial \lambda_s} \left( \frac{\partial \mathcal{L}_j}{\partial V_m} \right) & \frac{\partial}{\partial \lambda_{sj}} \left( \frac{\partial \mathcal{L}_j}{\partial V_m} \right) & \frac{\partial}{\partial \lambda_{Ls}} \left( \frac{\partial \mathcal{L}_j}{\partial V_m} \right) \\ \frac{\partial}{\partial \delta_s} \left( \frac{\partial \mathcal{L}_j}{\partial \lambda_m} \right) & \frac{\partial}{\partial V_s} \left( \frac{\partial \mathcal{L}_j}{\partial \lambda_m} \right) & \vdots & \ddots & \vdots \\ \frac{\partial}{\partial \delta_s} \left( \frac{\partial \mathcal{L}_j}{\partial \lambda_{mj}} \right) & \frac{\partial}{\partial V_s} \left( \frac{\partial \mathcal{L}_j}{\partial \lambda_{mj}} \right) & 0 & \dots & 0 \\ \frac{\partial}{\partial \delta_s} \left( \frac{\partial \mathcal{L}_j}{\partial \lambda_{Ls}} \right) & \frac{\partial}{\partial V_s} \left( \frac{\partial \mathcal{L}_j}{\partial \lambda_{Ls}} \right) & 0 & \dots & 0 \end{bmatrix} \quad 3.42$$

It must be noted that these new conditions don't only imply modifying the dimensionality of the matrices, but will also introduce changes to the equations used to calculate the derivatives with

respect to system states. This is due to the dependability of the newly added constraint on system voltages and angles.

## CHAPTER 4

### RESULTS AND DISCUSSION

#### 4.1 Introduction

A MATLAB prototype of the proposed scheme was examined on several test systems to validate its adequacy and efficiency. The following sections present a thorough illustration of the algorithm performance and compare its results against those produced using the continuation power flow of PSAT which is a MATLAB toolbox used for electric power system analysis and control.

It must be noted that the tolerance value was set to  $10^{-2}$  in all case studies investigated in this chapter.

#### 4.2 Testing the Scheme on the IEEE14-Bus System

The voltage collapse point of this system was found to be at loading parameter  $\lambda_{\max}$  equal to 4.0363. The algorithm took a total number of four iterations to arrive to this result. The step of intersection was chosen in the first two iterations whereas in the last two the algorithm selected the Newton step. No step rejections were issued in this case study.

Figure 4.1 shows the variations of the objective function through the iterative process. It can be noted that the objective function experiences a monotone decrease in its value which indicates the success of the Dog-leg scheme in selecting the proper step size and in controlling the radius of the trust region. The variations in this radius are depicted in figure 4.2. The increase of the

trust radius through the iterations is due to the fact that the reduction ratio was always above 0.75 for this case study.

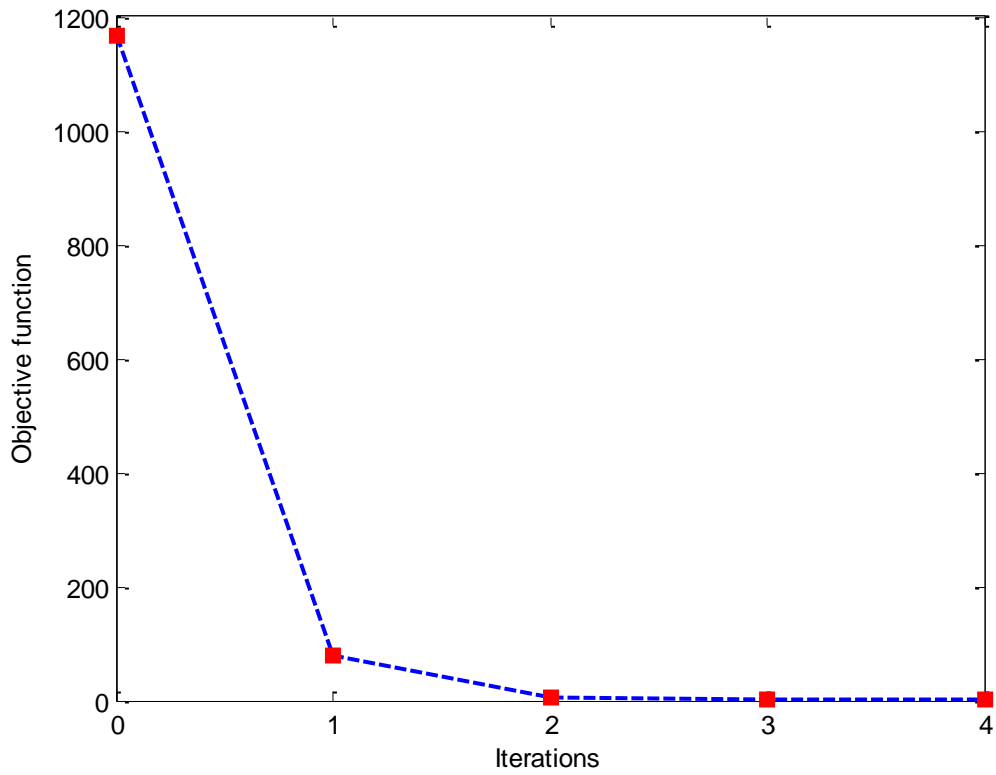


Figure 4.1 Variations of the Objective Function through the Iterations for the IEEE14-Bus System

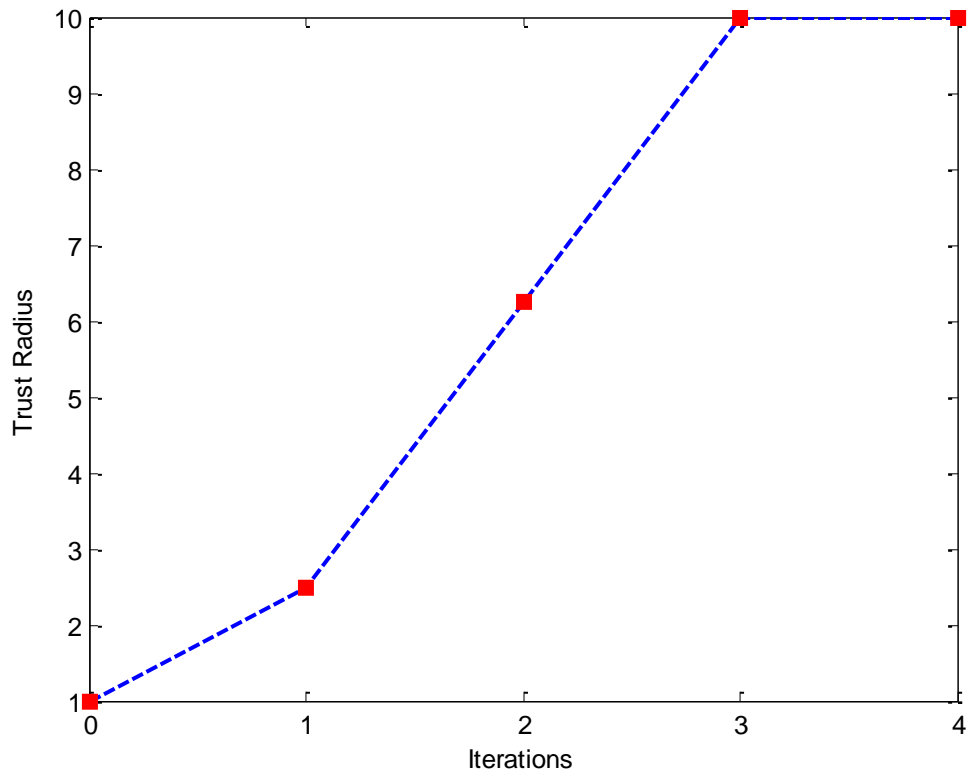


Figure 4.2 Trust Radius Dynamics for the IEEE14-Bus System

### 4.3 Testing the Scheme on the IEEE39-Bus System

Similar to the previous case study the scheme only took four iterations to converge. The collapse point was found to be at loading parameter equal to 2.3002. In this case, the step of intersection was used in the first iteration while the Newton step was selected in the remaining three. The changes of the trust radius through the four steps are similar to those depicted in figure 4.2. The variations of the objective function for this case are shown on figure 4.3.



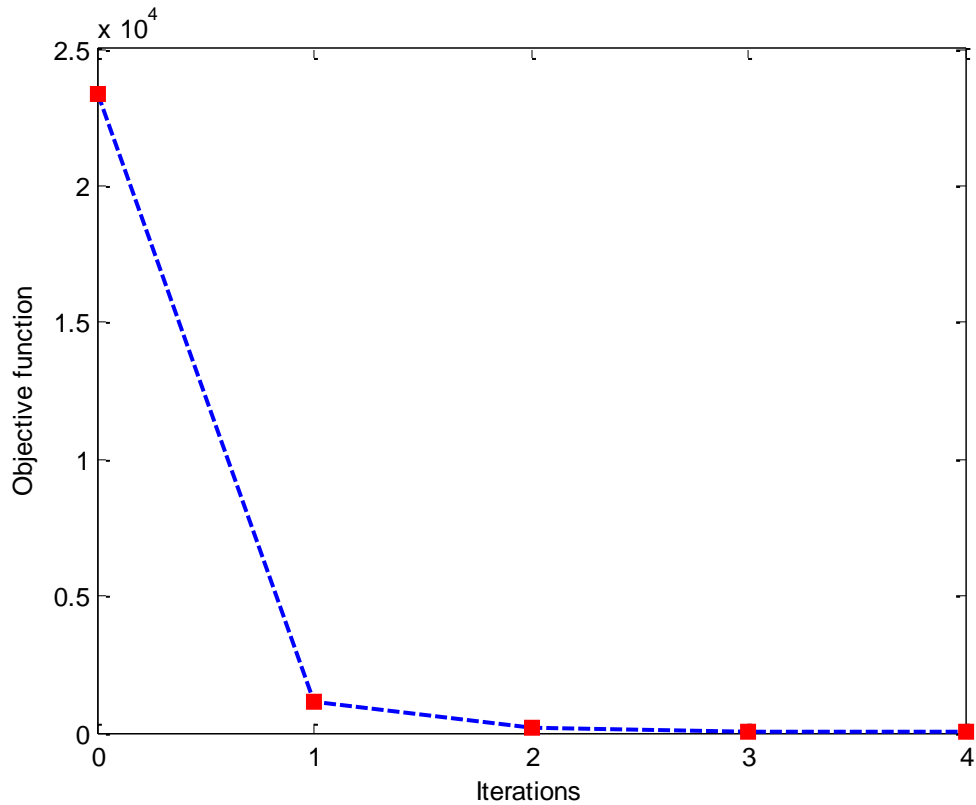


Figure 4.3 Variations of the Objective Function through the Iterations for the IEEE39-Bus System

#### 4.4 Testing the Scheme on the IEEE57-Bus System

When applied to the IEEE57-Bus system, the resulting loading parameter  $\lambda_{\max}$  was found to be 1.8465. To arrive at convergence, the algorithm took five iterations during which four step rejections were issued. This can be traced in figure 4.4 where the value of the objective function at the end of each iteration is shown. For instance, at the beginning of the second iteration the chosen step led to an increase in the objective function to a value of 5.8441. Consequently, this step was rejected and the trust radius was reduced from 2.5 to 0.625. A new Dog-leg trajectory was then formulated through the recalculation of the Cauchy point based on

the modified radius. This resulted in a new step which introduced a total reduction in the objective function. The algorithm then proceeded in a similar fashion through the remaining iterations until reaching a solution.

The dynamics that were experienced by the trust radius through the entire process are shown in figure 4.5. The algorithm was triggered to expand and shrink the radius several times due to the multiple step rejections encountered in this scenario.

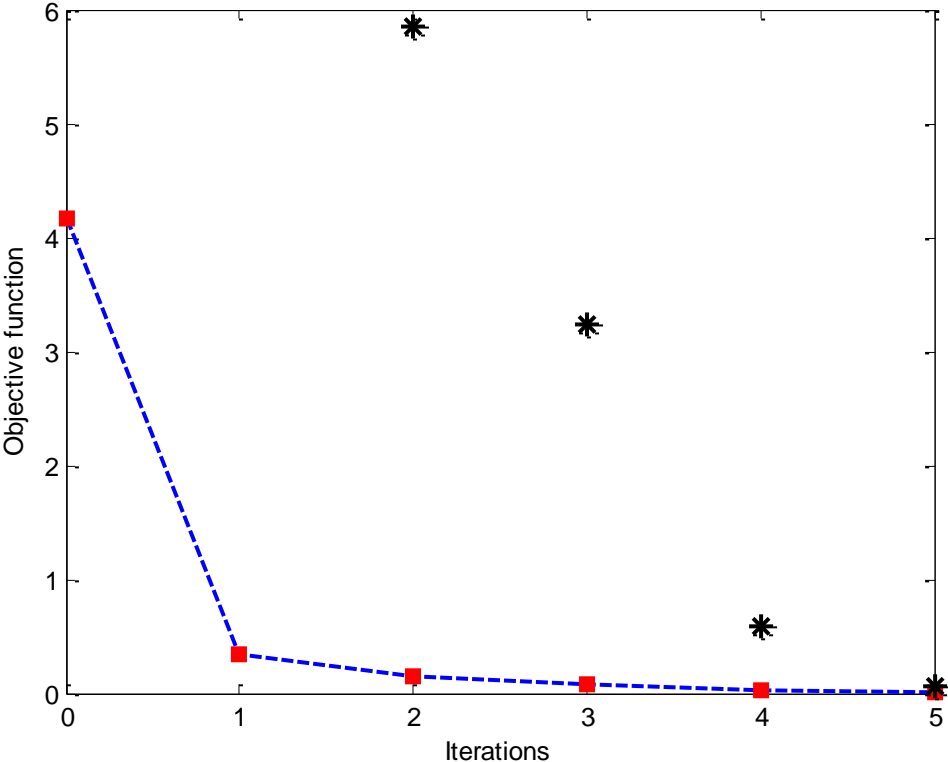


Figure 4.4 Variations of the Objective Function through the Iterations for the IEEE57-Bus System

The rejections are indicated by an asterisk (\*).

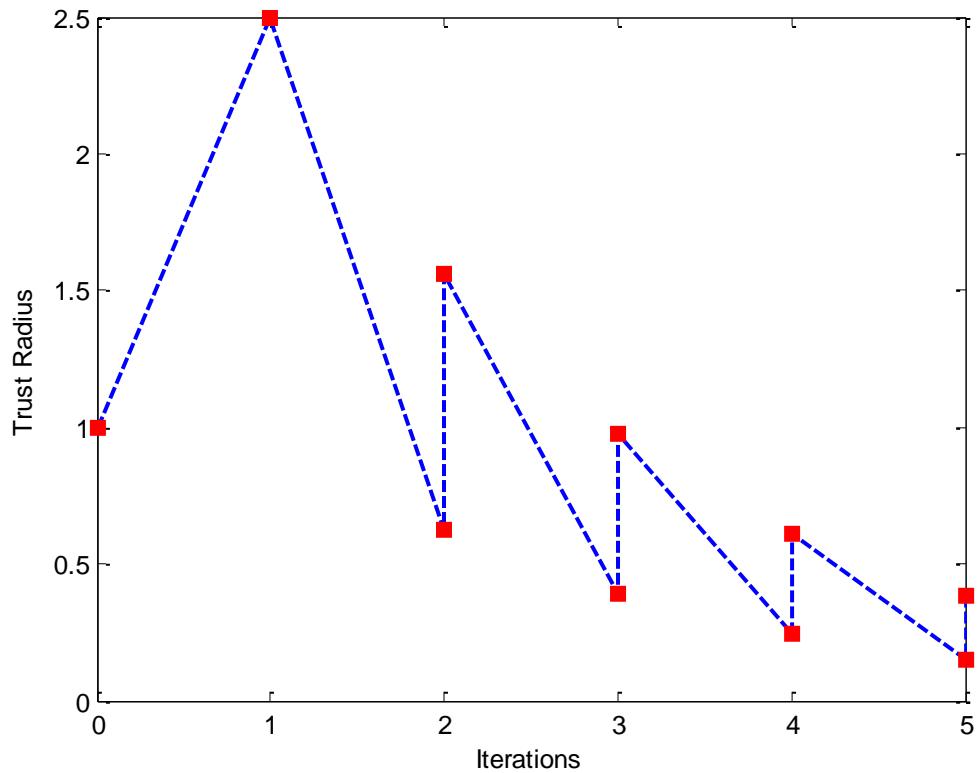


Figure 4.5 Trust Radius Dynamics for the IEEE57-Bus System

#### 4.5 Testing the Scheme on the IEEE118-Bus System

In this case, the scheme took 11 iterations to converge. The loading parameter  $\lambda_{\max}$  was found to be 3.1944. A single step rejection was issued as can be seen from figure 4.6 which depicts the variations of the objective function through the iterative process. Moreover, the variations experienced by the trust radius are shown in figure 4.7.

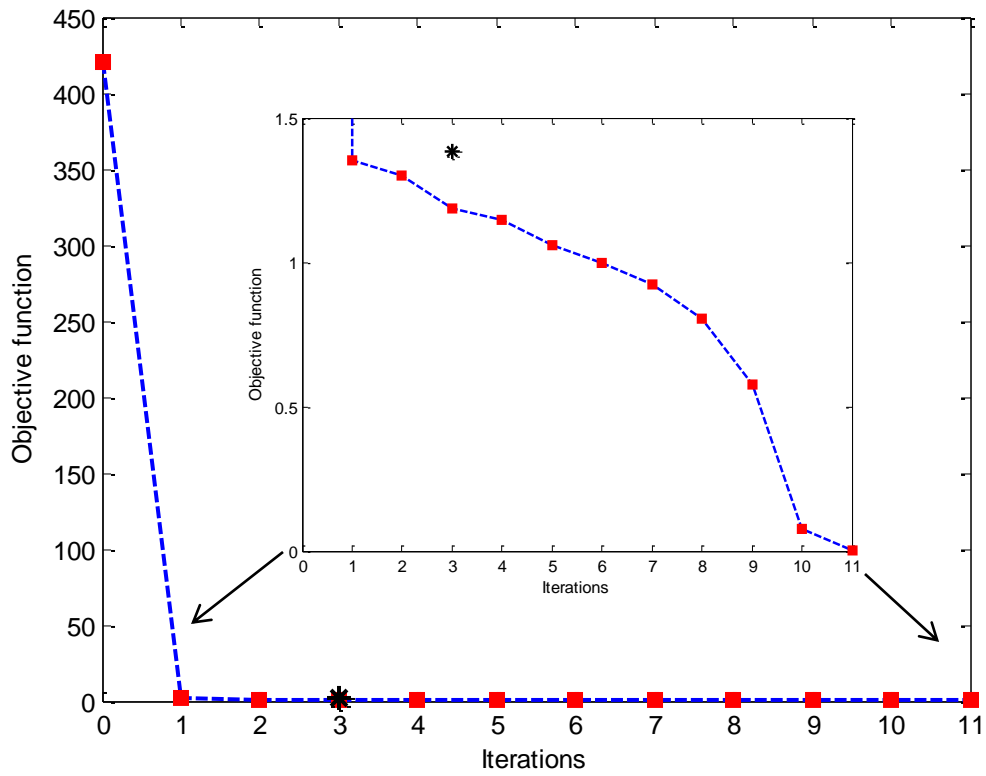


Figure 4.6 Variations of the Objective Function through the Iterations for the IEEE118-Bus System

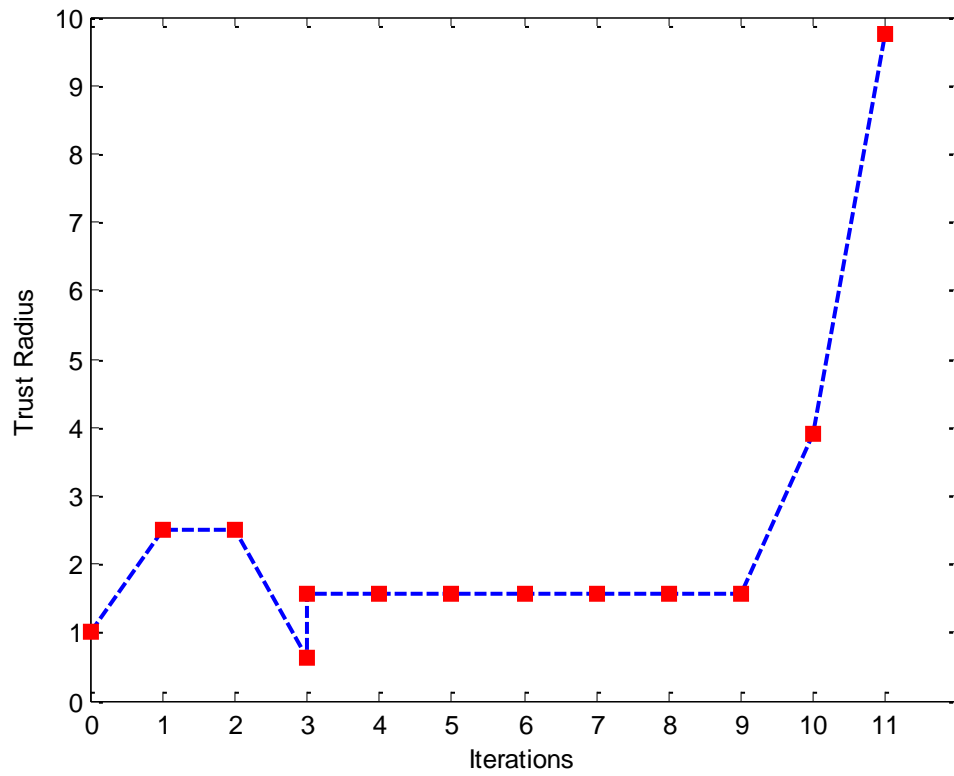


Figure 4.7 Trust Radius Dynamics for the IEEE118-Bus System

Table 4.1 summarizes the outcomes of the previous four case studies.

Table 4.1 Summary of the Suggested Method Performance

Test system	Loading Parameter ( $\lambda_{max}$ )	Number of iterations	Number of step rejections	Number of Cauchy steps	Number of Newton steps	Number of intersection steps
IEEE14	4.0363	4	0	0	2	2
IEEE39	2.3002	4	0	0	3	1
IEEE57	1.8465	5	4	0	3	6
IEEE118	3.1944	11	1	0	1	11

#### 4.6 Comparing the Performance of the Proposed Method against PSAT

Table 4.2 presents the results obtained using the MATLAB prototype of the proposed methodology along with those obtained using the predictor-corrector continuation power flow of PSAT. For reference, the solution times are obtained using an Intel(R) Xeon(R) CPU with 4GB of RAM running at 2.4GHz.

Table 4.2 Comparison between the Performances of the Suggested Algorithm and PSAT

Test system	Loading Parameter ( $\lambda_{max}$ )	$\lambda_{max}$ using proposed method	% Error in $\lambda_{max}$ using proposed method	Processing time of proposed method (sec)	$\lambda_{max}$ using PSAT	% Error in $\lambda_{max}$ using PSAT	Processing time of PSAT (sec)	Saving in time introduced by proposed method
IEEE14	4.04	4.0363	0.092	0.0156	4.0329	0.176	0.38058	95.9%
IEEE39	2.30	2.3002	0.0087	0.0312	2.3288	1.252	0.22734	86.2%
IEEE57	1.844	1.8465	0.136	0.0624	1.8358	0.445	1.1036	94.3%
IEEE118	3.19	3.1944	0.138	0.5616	3.1647	0.793	1.0443	46.22%

Based on table 4.2 it is evident that the suggested method is both faster and more accurate when compared to PSAT.

#### 4.7 Incorporation of Generator Reactive Power Limits

Table 4.3 summarizes the results obtained when forcing generator reactive power limits in the case of the IEEE14, IEEE57, and IEEE118-bus systems. It is to be noted that these result were obtained using an accuracy of 0.005.

Table 4.3 Maximum Loading Parameter When Including Generator Reactive Power Limits

Test system	Loading Parameter ( $\lambda_{max}$ )	No. of iterations	Number of step rejections	Processing Time (sec)	Total No. of generators in the System	No. of generators that hit reactive power limits
IEEE14	1.7647	5	2	0.0624	4	4
IEEE57	1.5694	4	2	0.5460	6	6
IEEE118	2.1467	7	1	3.5100	53	38

For the case of the IEEE14-bus system, the generated reactive power at P-V buses at calculated maximum loading parameter is shown in table 4.4 together with their maximum limits. It is evident that the algorithm succeeded in maintaining reactive power of generators within their limits. Tables 4.5 and 4.6 show that this is also true for the case of the IEEE57 and IEEE118, respectively.



Table 4.4 Generated Reactive Power at P-V Buses and Their Maximum Limits for the IEEE 14-Bus System

Bus No.	Reactive Power Limit (MVAR)	Generated Reactive Power at Maximum Loading Parameter (MVAR)
2	50	50.01
3	40	40.03
6	24	24.05
8	24	24

Table 4.5 Generated Reactive Power at P-V Buses and Their Maximum Limits for the IEEE 57-Bus System

Bus No.	Reactive Power Limit (MVAR)	Generated Reactive Power at Maximum Loading Parameter (MVAR)
2	50	48.51
3	60	60.28
6	25	24.99
8	200	199.97
9	9	9.09
12	155	154.83

Table 4.6 Generated Reactive Power at P-V Buses and Their Maximum Limits for the IEEE 118-Bus System

Bus No.	Reactive Power Limit (MVAR)	Generated Reactive Power at Maximum Loading Parameter (MVAR)	Bus No.	Reactive Power Limit (MVAR)	Generated Reactive Power at Maximum Loading Parameter (MVAR)
1	15	14.96	65	200	200
4	300	299.91	66	200	199.98
6	50	49.96	70	32	32
8	300	299.87	72	100	21.88
10	200	200	73	100	100
12	120	119.97	74	9	8.96
15	30	29.96	76	23	22.95
18	50	49.95	77	70	70.02
19	24	23.97	80	280	280.05
24	300	55.82	85	23	22.98
25	140	140.01	87	1000	44.14
26	1000	82.81	89	300	246.73
27	300	139.02	90	300	194.34

31	300	118.78	91	100	40.4
32	42	41.97	92	9	9
34	24	23.96	99	100	100
36	24	23.97	100	155	155.41
40	300	299.95	103	40	40.28
42	300	193.12	104	23	23.08
46	100	99.98	105	23	23.01
49	210	210.02	107	200	199.77
54	300	299.95	110	23	22.95
55	23	22.97	111	1000	60.9
56	15	14.98	112	1000	186.66
59	180	179.86	113	200	77.71
61	300	122.59	116	1000	217.97
62	20	19.98			

It is worth mentioning that when reactive power limits are imposed the PSAT toolbox took 0.57 seconds to converge in the case of the IEEE14-bus system whereas it generated error messages without producing a final result when testing both the IEEE57 and the IEEE118-bus systems.

## CHAPTER 5

### CONCLUSION

#### 5.1 Conclusion

In this work, the search for the maximum loading point of a power system is formulated as an optimization problem which is constrained by the rate of increase of active power at all system buses as well as the power factor relationship at load buses. The objective of this optimization along with the problem constraints are then used to formulate a Lagrangian function which is solved using the Dog-leg Trust Region scheme. This scheme is characterized by its robustness as it combines the reliability of the Steepest Descent method together with the fast convergence of the Newton line search.

The performance of the proposed methodology was examined using a variety of tests systems namely, the IEEE14, IEEE39, IEEE57 and IEEE118 bus systems. The algorithm succeeded in finding the maximum loading point of these systems without facing any convergence issues. Moreover, it was found that the suggested method provide a considerable saving in terms of processing time when compared to the continuation power flow analysis of PSAT.

The use of the Dog-leg Trust Region technique is what essentially makes the suggested algorithm powerful when compared to other direct method of maximum loading point calculations. This is primarily due to the employment of the Hessian matrix which doesn't suffer

from singularity issues at the voltage collapse point in contrary to the load flow Jacobian matrix deployed in other schemes. Furthermore, the problem of convergence dependability on initial conditions, faced when employing other optimization schemes that rely on Newton method, is eliminated in the proposed methodology. As a matter of fact, the compromise between the Cauchy and the Newton step along with the adjustable trust region radius serves as powerful safeguards that carefully guide the scheme to convergence.

Similar to other optimization methods, the incorporation of generator reactive power limits was easily made in the proposed scheme. However, what distinguishes this new method is the fact that generator reactive power constraints are dynamically added to the problem whenever a generator hits its limit. This is different from other methods that incorporate all the constraints from the beginning of the solution which results in a considerable increase in the dimensionality of the Hessian matrix. However, the proposed scheme eliminate this issue as reactive power constraints don't feature into the problem unless absolutely necessary.

## **5.2 Future Work**

The P-index [7] which is a voltage stability indicator that is based on normalized voltage and power sensitivities can be deployed with the proposed method to speed up the iterative process. This computationally inexpensive index provides an approximation of the voltage collapse point which can be used as a starting point by the developed scheme to reduce the number of iterations.

## REFERENCES

- [1] P. Kundur, N. J. Balu, and M. G. Lauby, *Power system stability and control*. McGraw-hill New York, 1994.
- [2] A. Jalali and M. Aldeen, "Novel continuation power flow algorithm," in *2016 IEEE International Conference on Power System Technology (POWERCON)*, 2016, pp. 1-6.
- [3] C. Gómez-Quiles, A. Gómez-Expósito, and W. Vargas, "Computation of Maximum Loading Points via the Factored Load Flow," *IEEE Transactions on Power Systems*, vol. 31, no. 5, pp. 4128-4134, 2016.
- [4] G. D. Irisarri, X. Wang, J. Tong, and S. Mokhtari, "Maximum loadability of power systems using interior point nonlinear optimization method," *IEEE Transactions on Power Systems*, vol. 12, no. 1, pp. 162-172, 1997.
- [5] A. F. Zeitune and R. de Souza Salgado, "A direct method to determine the maximum loadability bifurcation point in electric power systems," in *2nd International Conference on Engineering Optimization*, 2010.
- [6] V. Buljak, *Inverse analyses with model reduction: proper orthogonal decomposition in structural mechanics*. Springer Science & Business Media, 2011.
- [7] M. Kamel, "Development and Application of a New Voltage Stability Index for On-line Monitoring and Shedding," Master's thesis, The University of Tennessee at Chattanooga (UTC), 2016.

## APPENDIX A

### EQUATIONS FOR FORMING THE GRADIENT AND THE HESSIAN MATRICES

Assuming that  $\lambda_i$  is zero for all generator buses, since no constraint is imposed on the relationship between the active and reactive power of a generator bus, then equation (3.4) can be rewritten as:

$$\begin{aligned} \mathcal{L}_j = & P_j(V, \delta) + \sum_i \lambda_i (\beta_i P_i(V, \delta) - Q_i(V, \delta)) \\ & + \sum_{i \neq j} \lambda_{ij} (\alpha_{ij} P_j(V, \delta) - P_i(V, \delta)) \end{aligned} \quad \text{A.1}$$

Rearranging the terms of equation A.1 results in:

$$\begin{aligned} \mathcal{L}_j = & \left( 1 + \lambda_j \beta_j + \sum_{i \neq j} \lambda_{ij} \alpha_{ij} \right) P_j(V, \delta) - \lambda_j Q_j(V, \delta) \\ & + \sum_{i \neq j} (\lambda_i \beta_i - \lambda_{ij}) P_i(V, \delta) - \sum_{i \neq j} \lambda_i Q_i(V, \delta) \end{aligned} \quad \text{A.2}$$

The power flow equations at bus  $i$  can be expressed as shown in equations (A.3) and (A.4).

$$P_i = \sum_{k=1}^n Y_{ik} V_i V_k \cos(\theta_{ik} + \delta_k - \delta_i) \quad \text{A.3}$$

$$Q_i = - \sum_{k=1}^n Y_{ik} V_i V_k \sin(\theta_{ik} + \delta_k - \delta_i) \quad \text{A.4}$$



Substituting these expressions in equation (A.2):

$$\begin{aligned}
\mathcal{L}_j = & \left( 1 + \lambda_j \beta_j + \sum_{i \neq j} \lambda_{ij} \alpha_{ij} \right) \sum_{k=1}^n Y_{jk} V_j V_k \cos(\theta_{jk} + \delta_k - \delta_j) \\
& + \sum_{i \neq j} (\lambda_i \beta_i - \lambda_{ij}) \left( \sum_{k=1}^n Y_{ik} V_i V_k \cos(\theta_{ik} + \delta_k - \delta_i) \right) \\
& + \sum_{i \neq j} \lambda_i \left( \sum_{k=1}^n Y_{ik} V_i V_k \sin(\theta_{ik} + \delta_k - \delta_i) \right) \\
& + \lambda_j \sum_{k=1}^n Y_{jk} V_j V_k \sin(\theta_{jk} + \delta_k - \delta_j)
\end{aligned} \tag{A.5}$$

Equations (A.6) through (A.11) show the first derivatives of equation (A.5) with respect to all independent variables. These derivatives are used to construct the gradient matrix  $\nabla f(x_k)$  encountered in equation (3.22).

$$\begin{aligned}
\left. \frac{\partial \mathcal{L}_j}{\partial \delta_m} \right|_{m \neq j} = & \left( 1 + \lambda_j \beta_j + \sum_{i \neq j} \lambda_{ij} \alpha_{ij} \right) * (-Y_{jm} V_j V_m \sin(\theta_{jm} + \delta_m - \delta_j)) \\
& + \lambda_m (-P_m + Y_{mm} V_m^2 \cos(\theta_{mm})) + \sum_{\substack{i \neq j \\ i \neq m}} \lambda_i * Y_{im} V_i V_m \cos(\theta_{im} + \delta_m - \delta_i) \\
& + \sum_{\substack{i \neq j \\ i \neq m}} (\lambda_i \beta_i - \lambda_{ij}) * (-Y_{im} V_i V_m \sin(\theta_{im} + \delta_m - \delta_i)) \\
& + (\lambda_m \beta_m - \lambda_{mj}) * (-Q_m - Y_{mm} V_m^2 \sin(\theta_{mm})) \\
& + \lambda_j Y_{jm} V_j V_m \cos(\theta_{jm} + \delta_m - \delta_j)
\end{aligned} \tag{A.6}$$

$$\begin{aligned}
\left. \frac{\partial \mathcal{L}_j}{\partial \delta_m} \right|_{m=j} &= \left( 1 + \lambda_j \beta_j + \sum_{i \neq j} \lambda_{ij} \alpha_{ij} \right) * (-Q_j - Y_{jj} V_j^2 \sin(\theta_{jj})) \\
&+ \sum_{i \neq j} \lambda_i Y_{ij} V_i V_j \cos(\theta_{ij} + \delta_j - \delta_i) + \lambda_j (-P_j + Y_{jj} V_j^2 \cos(\theta_{jj})) \\
&+ \sum_{i \neq j} (\lambda_i \beta_i - \lambda_{ij}) * (-Y_{ij} V_i V_j \sin(\theta_{ij} + \delta_j - \delta_i))
\end{aligned} \tag{A.7}$$

$$\begin{aligned}
\left. \frac{\partial \mathcal{L}_j}{\partial V_m} \right|_{m \neq j} &= \left( 1 + \lambda_j \beta_j + \sum_{i \neq j} \lambda_{ij} \alpha_{ij} \right) * Y_{jm} V_j \cos(\theta_{jm} + \delta_m - \delta_j) \\
&+ \lambda_m \left( -\frac{Q_m}{V_m} + Y_{mm} V_m \sin(\theta_{mm}) \right) + \lambda_j Y_{jm} V_j \sin(\theta_{jm} + \delta_m - \delta_j) \\
&+ \sum_{\substack{i \neq j \\ i \neq m}} (\lambda_i \beta_i - \lambda_{ij}) * (Y_{im} V_i \cos(\theta_{im} + \delta_m - \delta_i)) \\
&+ (\lambda_m \beta_m - \lambda_{mj}) * \left( \frac{P_m}{V_m} + Y_{mm} V_m \cos(\theta_{mm}) \right) \\
&+ \sum_{\substack{i \neq j \\ i \neq m}} \lambda_i Y_{im} V_i \sin(\theta_{im} + \delta_m - \delta_i)
\end{aligned} \tag{A.8}$$

$$\begin{aligned}
\left. \frac{\partial \mathcal{L}_j}{\partial V_m} \right|_{m=j} &= \left( 1 + \lambda_j \beta_j + \sum_{i \neq j} \lambda_{ij} \alpha_{ij} \right) * \left( \frac{P_j}{V_j} + Y_{jj} V_j \cos(\theta_{jj}) \right) \\
&+ \sum_{i \neq j} (\lambda_i \beta_i - \lambda_{ij}) Y_{ij} V_i \cos(\theta_{ij} + \delta_j - \delta_i) + \sum_{i \neq j} \lambda_i Y_{ij} V_i \sin(\theta_{ij} + \delta_j - \delta_i)
\end{aligned} \tag{A.9}$$

$$+\lambda_j \left( -\frac{Q_j}{V_j} + Y_{jj} V_j \sin(\theta_{jj}) \right)$$

$$\frac{\partial \mathcal{L}_j}{\partial \lambda_m} = \beta_m P_m - Q_m \quad \text{A.10}$$

$$\frac{\partial \mathcal{L}_j}{\partial \lambda_{mj}} = \alpha_{mj} P_j - P_m \quad \text{A.11}$$

Equations (A.12) through (A.61) show expressions of the second derivatives of equation (A.5) with respect to all independent variables. These derivatives are used to construct the Hessian matrix  $\nabla^2 f(x_k)$  shown in equation (3.26).

$$\begin{aligned} \left. \frac{\partial}{\partial \delta_s} \left( \frac{\partial \mathcal{L}_j}{\partial \delta_m} \right) \right|_{s \neq m \neq j} &= (\lambda_s \beta_s - \lambda_{sj}) Y_{sm} V_s V_m \cos(\theta_{sm} + \delta_m - \delta_s) \\ &+ \lambda_s Y_{sm} V_s V_m \sin(\theta_{sm} + \delta_m - \delta_s) + \lambda_m Y_{ms} V_s V_m \sin(\theta_{ms} + \delta_s - \delta_m) \\ &+ (\lambda_m \beta_m - \lambda_{mj}) Y_{ms} V_s V_m \cos(\theta_{ms} + \delta_s - \delta_m) \end{aligned} \quad \text{A.12}$$

$$\begin{aligned} \left. \frac{\partial}{\partial \delta_s} \left( \frac{\partial \mathcal{L}_j}{\partial \delta_m} \right) \right|_{s=m \neq j} &= \lambda_s (Q_s + Y_{ss} V_s^2 \sin(\theta_{ss})) - \lambda_j Y_{js} V_j V_s \sin(\theta_{js} + \delta_s - \delta_j) \\ &- \sum_{\substack{i \neq j \\ i \neq s}} \lambda_i Y_{is} V_i V_s \sin(\theta_{is} + \delta_s - \delta_i) + (\lambda_s \beta_s - \lambda_{sj}) * (-P_s + Y_{ss} V_s^2 \cos(\theta_{ss})) \end{aligned} \quad \text{A.13}$$

$$\begin{aligned}
& + \left( 1 + \lambda_j \beta_j + \sum_{i \neq j} \lambda_{ij} \alpha_{ij} \right) * (-Y_{js} V_j V_s \cos(\theta_{js} + \delta_s - \delta_j)) \\
& - \sum_{\substack{i \neq j \\ i \neq s}} (\lambda_i \beta_i - \lambda_{ij}) Y_{is} V_i V_s \cos(\theta_{is} + \delta_s - \delta_i)
\end{aligned}$$

$$\left. \frac{\partial}{\partial \delta_s} \left( \frac{\partial \mathcal{L}_j}{\partial \delta_m} \right) \right|_{s=j \neq m} = (\lambda_m \beta_m - \lambda_{mj}) * (Y_{mj} V_m V_j \cos(\theta_{mj} + \delta_j - \delta_m))$$

$$+ \lambda_j Y_{jm} V_j V_m \sin(\theta_{jm} + \delta_m - \delta_j) + \lambda_m Y_{mj} V_m V_j \sin(\theta_{mj} + \delta_j - \delta_m)$$

A.14

$$+ \left( 1 + \lambda_j \beta_j + \sum_{i \neq j} \lambda_{ij} \alpha_{ij} \right) * (Y_{jm} V_j V_m \cos(\theta_{jm} + \delta_m - \delta_j))$$

$$\left. \frac{\partial}{\partial \delta_s} \left( \frac{\partial \mathcal{L}_j}{\partial \delta_m} \right) \right|_{s \neq j=m} = \left( 1 + \lambda_j \beta_j + \sum_{i \neq j} \lambda_{ij} \alpha_{ij} \right) * (Y_{js} V_j V_s \cos(\theta_{js} + \delta_s - \delta_j))$$

A.15

$$+ \lambda_j Y_{js} V_j V_s \sin(\theta_{js} + \delta_s - \delta_j) + \lambda_s Y_{sj} V_s V_j \sin(\theta_{sj} + \delta_j - \delta_s)$$

$$+ (\lambda_s \beta_s - \lambda_{sj}) * (Y_{sj} V_s V_j \cos(\theta_{sj} + \delta_j - \delta_s))$$

$$\left. \frac{\partial}{\partial \delta_s} \left( \frac{\partial \mathcal{L}_j}{\partial \delta_m} \right) \right|_{s=j=m} = \left( 1 + \lambda_j \beta_j + \sum_{i \neq j} \lambda_{ij} \alpha_{ij} \right) * (-P_j + Y_{jj} V_j^2 \cos(\theta_{jj}))$$

A.16

$$+ \lambda_j (Q_j + Y_{jj} V_j^2 \sin(\theta_{jj})) - \sum_{i \neq j} (\lambda_i \beta_i - \lambda_{ij}) Y_{ij} V_i V_j \cos(\theta_{ij} + \delta_j - \delta_i)$$

$$-\sum_{i \neq j} \lambda_i Y_{ij} V_i V_j \sin(\theta_{ij} + \delta_j - \delta_i)$$

$$\left. \frac{\partial}{\partial V_s} \left( \frac{\partial \mathcal{L}_j}{\partial \delta_m} \right) \right|_{s \neq j \neq m} = -(\lambda_s \beta_s - \lambda_{sj}) Y_{sm} V_m \sin(\theta_{sm} + \delta_m - \delta_s)$$

$$+(\lambda_m \beta_m - \lambda_{mj}) Y_{ms} V_m \sin(\theta_{ms} + \delta_s - \delta_m) + \lambda_s Y_{sm} V_m \cos(\theta_{sm} + \delta_m - \delta_s) \\ - \lambda_m Y_{ms} V_m \cos(\theta_{ms} + \delta_s - \delta_m)$$

A.17

$$\left. \frac{\partial}{\partial V_s} \left( \frac{\partial \mathcal{L}_j}{\partial \delta_m} \right) \right|_{s=m \neq j} = \left( 1 + \lambda_j \beta_j + \sum_{i \neq j} \lambda_{ij} \alpha_{ij} \right) * (-Y_{js} V_j \sin(\theta_{js} + \delta_s - \delta_j))$$

$$+ \lambda_j Y_{js} V_j \cos(\theta_{js} + \delta_s - \delta_j) - \sum_{\substack{i \neq j \\ i \neq s}} (\lambda_i \beta_i - \lambda_{ij}) Y_{is} V_i \sin(\theta_{is} + \delta_s - \delta_i)$$

A.18

$$+ \sum_{\substack{i \neq j \\ i \neq s}} \lambda_i Y_{is} V_i \cos(\theta_{is} + \delta_s - \delta_i) + (\lambda_s \beta_s - \lambda_{sj}) * \left( -\frac{Q_s}{V_s} - Y_{ss} V_s \sin(\theta_{ss}) \right)$$

$$+ \lambda_s \left( -\frac{P_s}{V_s} + Y_{ss} V_s \cos(\theta_{ss}) \right)$$

$$\left. \frac{\partial}{\partial V_s} \left( \frac{\partial \mathcal{L}_j}{\partial \delta_m} \right) \right|_{s=j \neq m} = (\lambda_m \beta_m - \lambda_{mj}) * (Y_{mj} V_m \sin(\theta_{mj} + \delta_j - \delta_m))$$

$$+ \lambda_j Y_{jm} V_m \cos(\theta_{jm} + \delta_m - \delta_j) - \lambda_m Y_{mj} V_m \cos(\theta_{mj} + \delta_j - \delta_m)$$

A.19

$$+ \left( 1 + \lambda_j \beta_j + \sum_{i \neq j} \lambda_{ij} \alpha_{ij} \right) * (-Y_{jm} V_m \sin(\theta_{jm} + \delta_m - \delta_j))$$

$$\begin{aligned}
\left. \frac{\partial}{\partial V_s} \left( \frac{\partial \mathcal{L}_j}{\partial \delta_m} \right) \right|_{s \neq j=m} &= \left( 1 + \lambda_j \beta_j + \sum_{i \neq j} \lambda_{ij} \alpha_{ij} \right) * (Y_{js} V_j \sin(\theta_{js} + \delta_s - \delta_j)) \\
&- \lambda_j Y_{js} V_j \cos(\theta_{js} + \delta_s - \delta_j) - (\lambda_s \beta_s - \lambda_{sj}) * (Y_{sj} V_j \sin(\theta_{sj} + \delta_j - \delta_s)) \\
&+ \lambda_s Y_{sj} V_j \cos(\theta_{sj} + \delta_j - \delta_s)
\end{aligned} \tag{A.20}$$

$$\begin{aligned}
\left. \frac{\partial}{\partial V_s} \left( \frac{\partial \mathcal{L}_j}{\partial \delta_m} \right) \right|_{s=j=m} &= \left( 1 + \lambda_j \beta_j + \sum_{i \neq j} \lambda_{ij} \alpha_{ij} \right) * \left( -\frac{Q_j}{V_j} - Y_{jj} V_j \sin(\theta_{jj}) \right) \\
&+ \lambda_j \left( -\frac{P_j}{V_j} + Y_{jj} V_j \cos(\theta_{jj}) \right) - \sum_{i \neq j} (\lambda_i \beta_i - \lambda_{ij}) Y_{ij} V_i \sin(\theta_{ij} + \delta_j - \delta_i) \\
&+ \sum_{i \neq j} \lambda_i Y_{ij} V_i \cos(\theta_{ij} + \delta_j - \delta_i)
\end{aligned} \tag{A.21}$$

$$\begin{aligned}
\left. \frac{\partial}{\partial \lambda_s} \left( \frac{\partial \mathcal{L}_j}{\partial \delta_m} \right) \right|_{s \neq j \neq m} &= -\beta_s Y_{sm} V_s V_m \sin(\theta_{sm} + \delta_m - \delta_s) \\
&+ Y_{sm} V_s V_m \cos(\theta_{sm} + \delta_m - \delta_s)
\end{aligned} \tag{A.22}$$

$$\left. \frac{\partial}{\partial \lambda_s} \left( \frac{\partial \mathcal{L}_j}{\partial \delta_m} \right) \right|_{s=m \neq j} = \beta_s (-Q_s - Y_{ss} V_s^2 \sin(\theta_{ss})) - P_s + Y_{ss} V_s^2 \cos(\theta_{ss}) \tag{A.23}$$

$$\left. \frac{\partial}{\partial \lambda_s} \left( \frac{\partial \mathcal{L}_j}{\partial \delta_m} \right) \right|_{s=j \neq m} = -\beta_j Y_{jm} V_j V_m \sin(\theta_{jm} + \delta_m - \delta_j) \tag{A.24}$$

$$+Y_{jm}V_jV_m \cos(\theta_{jm} + \delta_m - \delta_j)$$

$$\left. \frac{\partial}{\partial \lambda_s} \left( \frac{\partial \mathcal{L}_j}{\partial \delta_m} \right) \right|_{s \neq j=m} = -\beta_s Y_{sj} V_s V_j \sin(\theta_{sj} + \delta_j - \delta_s)$$
A.25

$$+Y_{sj} V_s V_j \cos(\theta_{sj} + \delta_j - \delta_s)$$

$$\left. \frac{\partial}{\partial \lambda_s} \left( \frac{\partial \mathcal{L}_j}{\partial \delta_m} \right) \right|_{s=j=m} = \beta_j (-Q_j - Y_{jj} V_j^2 \sin(\theta_{jj})) - P_j + Y_{jj} V_j^2 \cos(\theta_{jj})$$
A.26

$$\left. \frac{\partial}{\partial \lambda_{sj}} \left( \frac{\partial \mathcal{L}_j}{\partial \delta_m} \right) \right|_{s \neq j \neq m} = -\alpha_{sj} Y_{jm} V_j V_m \sin(\theta_{jm} + \delta_m - \delta_j)$$
A.27

$$-Y_{sm} V_s V_m \sin(\theta_{sm} + \delta_m - \delta_s)$$

$$\left. \frac{\partial}{\partial \lambda_{sj}} \left( \frac{\partial \mathcal{L}_j}{\partial \delta_m} \right) \right|_{s=m \neq j} = -\alpha_{sj} Y_{js} V_j V_s \sin(\theta_{js} + \delta_s - \delta_j) + Q_s + Y_{ss} V_s^2 \sin(\theta_{ss})$$
A.28

$$\left. \frac{\partial}{\partial \lambda_{sj}} \left( \frac{\partial \mathcal{L}_j}{\partial \delta_m} \right) \right|_{s \neq m=j} = \alpha_{sj} (-Q_j - Y_{jj} V_j^2 \sin(\theta_{jj}))$$
A.29

$$+Y_{sj} V_s V_j \sin(\theta_{sj} + \delta_j - \delta_s)$$

$$\left. \frac{\partial}{\partial \delta_s} \left( \frac{\partial \mathcal{L}_j}{\partial V_m} \right) \right|_{s \neq j \neq m} = (\lambda_s \beta_s - \lambda_{sj}) Y_{sm} V_s \sin(\theta_{sm} + \delta_m - \delta_s)$$
A.30

$$\begin{aligned}
& -(\lambda_m \beta_m - \lambda_{mj}) Y_{ms} V_s \sin(\theta_{ms} + \delta_s - \delta_m) - \lambda_s Y_{sm} V_s \cos(\theta_{sm} + \delta_m - \delta_s) \\
& \quad + \lambda_m Y_{ms} V_s \cos(\theta_{ms} + \delta_s - \delta_m)
\end{aligned}$$

$$\begin{aligned}
\frac{\partial}{\partial \delta_s} \left( \frac{\partial \mathcal{L}_j}{\partial V_m} \right) \Big|_{s=m \neq j} &= \left( 1 + \lambda_j \beta_j + \sum_{i \neq j} \lambda_{ij} \alpha_{ij} \right) * (-Y_{js} V_j \sin(\theta_{js} + \delta_s - \delta_j)) \\
& + \lambda_j Y_{js} V_j \cos(\theta_{js} + \delta_s - \delta_j) - \sum_{\substack{i \neq j \\ i \neq s}} (\lambda_i \beta_i - \lambda_{ij}) Y_{is} V_i \sin(\theta_{is} + \delta_s - \delta_i)
\end{aligned}$$

A.31

$$\begin{aligned}
& + (\lambda_s \beta_s - \lambda_{sj}) * \left( -\frac{Q_s}{V_s} - Y_{ss} V_s \sin(\theta_{ss}) \right) + \sum_{\substack{i \neq j \\ i \neq s}} \lambda_i Y_{is} V_i \cos(\theta_{is} + \delta_s - \delta_i) \\
& \quad + \lambda_s \left( -\frac{P_s}{V_s} + Y_{ss} V_s \cos(\theta_{ss}) \right)
\end{aligned}$$

$$\frac{\partial}{\partial \delta_s} \left( \frac{\partial \mathcal{L}_j}{\partial V_m} \right) \Big|_{s=j \neq m} = \left( 1 + \lambda_j \beta_j + \sum_{i \neq j} \lambda_{ij} \alpha_{ij} \right) * (Y_{jm} V_j \sin(\theta_{jm} + \delta_m - \delta_j))$$

A.32

$$\begin{aligned}
& -\lambda_j Y_{jm} V_j \cos(\theta_{jm} + \delta_m - \delta_j) + \lambda_m Y_{mj} V_j \cos(\theta_{mj} + \delta_j - \delta_m) \\
& \quad - (\lambda_m \beta_m - \lambda_{mj}) * (Y_{mj} V_j \sin(\theta_{mj} + \delta_j - \delta_m))
\end{aligned}$$

$$\frac{\partial}{\partial \delta_s} \left( \frac{\partial \mathcal{L}_j}{\partial V_m} \right) \Big|_{s \neq j = m} = \left( 1 + \lambda_j \beta_j + \sum_{i \neq j} \lambda_{ij} \alpha_{ij} \right) * (-Y_{js} V_s \sin(\theta_{js} + \delta_s - \delta_j))$$

A.33

$$\begin{aligned}
& + \lambda_j Y_{js} V_s \cos(\theta_{js} + \delta_s - \delta_j) + (\lambda_s \beta_s - \lambda_{sj}) * (Y_{sj} V_s \sin(\theta_{sj} + \delta_j - \delta_s)) \\
& \quad - \lambda_s Y_{sj} V_s \cos(\theta_{sj} + \delta_j - \delta_s)
\end{aligned}$$



$$\begin{aligned}
\left. \frac{\partial}{\partial \delta_s} \left( \frac{\partial \mathcal{L}_j}{\partial V_m} \right) \right|_{s=m=j} &= \left( 1 + \lambda_j \beta_j + \sum_{i \neq j} \lambda_{ij} \alpha_{ij} \right) * \left( -\frac{Q_j}{V_j} - Y_{jj} V_j \sin(\theta_{jj}) \right) \\
&+ \lambda_j \left( -\frac{P_j}{V_j} + Y_{jj} V_j \cos(\theta_{jj}) \right) - \sum_{i \neq j} (\lambda_i \beta_i - \lambda_{ij}) Y_{ij} V_i \sin(\theta_{ij} + \delta_j - \delta_i) \\
&\quad + \sum_{i \neq j} \lambda_i Y_{ij} V_i \cos(\theta_{ij} + \delta_j - \delta_i)
\end{aligned} \tag{A.34}$$

$$\begin{aligned}
\left. \frac{\partial}{\partial V_s} \left( \frac{\partial \mathcal{L}_j}{\partial V_m} \right) \right|_{s \neq j \neq m} &= (\lambda_s \beta_s - \lambda_{sj}) Y_{sm} \cos(\theta_{sm} + \delta_m - \delta_s) \\
&+ (\lambda_m \beta_m - \lambda_{mj}) Y_{ms} \cos(\theta_{ms} + \delta_s - \delta_m) + \lambda_s Y_{sm} \sin(\theta_{sm} + \delta_m - \delta_s) \\
&\quad + \lambda_m Y_{ms} \sin(\theta_{ms} + \delta_s - \delta_m)
\end{aligned} \tag{A.35}$$

$$\left. \frac{\partial}{\partial V_s} \left( \frac{\partial \mathcal{L}_j}{\partial V_m} \right) \right|_{s=m \neq j} = 2(\lambda_s \beta_s - \lambda_{sj}) Y_{ss} \cos(\theta_{ss}) + 2\lambda_s Y_{ss} \sin(\theta_{ss}) \tag{A.36}$$

$$\begin{aligned}
\left. \frac{\partial}{\partial V_s} \left( \frac{\partial \mathcal{L}_j}{\partial V_m} \right) \right|_{s=j \neq m} &= \left( 1 + \lambda_j \beta_j + \sum_{i \neq j} \lambda_{ij} \alpha_{ij} \right) * (Y_{jm} \cos(\theta_{jm} + \delta_m - \delta_j)) \\
&+ \lambda_j Y_{jm} \sin(\theta_{jm} + \delta_m - \delta_j) + (\lambda_m \beta_m - \lambda_{mj}) * (Y_{mj} \cos(\theta_{mj} + \delta_j - \delta_m)) \\
&\quad + \lambda_m Y_{mj} \sin(\theta_{mj} + \delta_j - \delta_m)
\end{aligned} \tag{A.37}$$

$$\begin{aligned}
\left. \frac{\partial}{\partial V_s} \left( \frac{\partial \mathcal{L}_j}{\partial V_m} \right) \right|_{s \neq j = m} &= \left( 1 + \lambda_j \beta_j + \sum_{i \neq j} \lambda_{ij} \alpha_{ij} \right) * (Y_{js} \cos(\theta_{js} + \delta_s - \delta_j)) \\
&+ \lambda_j Y_{js} \sin(\theta_{js} + \delta_s - \delta_j) + (\lambda_s \beta_s - \lambda_{sj}) * (Y_{sj} \cos(\theta_{sj} + \delta_j - \delta_s))
\end{aligned} \tag{A.38}$$

$$+\lambda_s Y_{sj} \sin(\theta_{sj} + \delta_j - \delta_s)$$

$$\begin{aligned} \frac{\partial}{\partial V_s} \left( \frac{\partial \mathcal{L}_j}{\partial V_m} \right) \Big|_{s=j=m} &= \left( 1 + \lambda_j \beta_j + \sum_{i \neq j} \lambda_{ij} \alpha_{ij} \right) * (2Y_{jj} \cos(\theta_{jj})) \\ &+ 2\lambda_j Y_{jj} \sin(\theta_{jj}) \end{aligned} \quad \text{A.39}$$

$$\begin{aligned} \frac{\partial}{\partial \lambda_s} \left( \frac{\partial \mathcal{L}_j}{\partial V_m} \right) \Big|_{s \neq j \neq m} &= \beta_s Y_{sm} V_s \cos(\theta_{sm} + \delta_m - \delta_s) \\ &+ Y_{sm} V_s \sin(\theta_{sm} + \delta_m - \delta_s) \end{aligned} \quad \text{A.40}$$

$$\frac{\partial}{\partial \lambda_s} \left( \frac{\partial \mathcal{L}_j}{\partial V_m} \right) \Big|_{s=m \neq j} = \beta_s \left( \frac{P_s}{V_s} + Y_{ss} V_s \cos(\theta_{ss}) \right) - \frac{Q_s}{V_s} + Y_{ss} V_s \sin(\theta_{ss}) \quad \text{A.41}$$

$$\begin{aligned} \frac{\partial}{\partial \lambda_s} \left( \frac{\partial \mathcal{L}_j}{\partial V_m} \right) \Big|_{s=j \neq m} &= \beta_j Y_{jm} V_j \cos(\theta_{jm} + \delta_m - \delta_j) \\ &+ Y_{jm} V_j \sin(\theta_{jm} + \delta_m - \delta_j) \end{aligned} \quad \text{A.42}$$

$$\frac{\partial}{\partial \lambda_s} \left( \frac{\partial \mathcal{L}_j}{\partial V_m} \right) \Big|_{s \neq j=m} = \beta_s Y_{sj} V_s \cos(\theta_{sj} + \delta_j - \delta_s) + Y_{sj} V_s \sin(\theta_{sj} + \delta_j - \delta_s) \quad \text{A.43}$$

$$\frac{\partial}{\partial \lambda_s} \left( \frac{\partial \mathcal{L}_j}{\partial V_m} \right) \Big|_{s=m=j} = \beta_j \left( \frac{P_j}{V_j} + Y_{jj} V_j \cos(\theta_{jj}) \right) - \frac{Q_j}{V_j} + Y_{jj} V_j \sin(\theta_{jj}) \quad \text{A.44}$$

$$\begin{aligned} \left. \frac{\partial}{\partial \lambda_{sj}} \left( \frac{\partial \mathcal{L}_j}{\partial V_m} \right) \right|_{s \neq m \neq j} &= \alpha_{sj} Y_{jm} V_j \cos(\theta_{jm} + \delta_m - \delta_j) \\ &\quad - Y_{sm} V_s \cos(\theta_{sm} + \delta_m - \delta_s) \end{aligned} \quad \text{A.45}$$

$$\left. \frac{\partial}{\partial \lambda_{sj}} \left( \frac{\partial \mathcal{L}_j}{\partial V_m} \right) \right|_{s=m \neq j} = \alpha_{sj} Y_{js} V_j \cos(\theta_{js} + \delta_s - \delta_j) - \frac{P_s}{V_s} - Y_{ss} V_s \cos(\theta_{ss}) \quad \text{A.46}$$

$$\left. \frac{\partial}{\partial \lambda_{sj}} \left( \frac{\partial \mathcal{L}_j}{\partial V_m} \right) \right|_{s \neq j=m} = \alpha_{sj} \left( \frac{P_j}{V_j} + Y_{jj} V_j \cos(\theta_{jj}) \right) - Y_{sj} V_s \cos(\theta_{sj} + \delta_j - \delta_s) \quad \text{A.47}$$

$$\begin{aligned} \left. \frac{\partial}{\partial \delta_s} \left( \frac{\partial \mathcal{L}_j}{\partial \lambda_m} \right) \right|_{s \neq m} &= -\beta_m Y_{ms} V_m V_s \sin(\theta_{ms} + \delta_s - \delta_m) \\ &\quad + Y_{ms} V_m V_s \cos(\theta_{ms} + \delta_s - \delta_s) \end{aligned} \quad \text{A.48}$$

$$\left. \frac{\partial}{\partial \delta_s} \left( \frac{\partial \mathcal{L}_j}{\partial \lambda_m} \right) \right|_{s=m} = \beta_m (-Q_s - Y_{ss} V_s^2 \sin(\theta_{ss})) - P_s + Y_{ss} V_s^2 \cos(\theta_{ss}) \quad \text{A.49}$$

$$\begin{aligned} \left. \frac{\partial}{\partial V_s} \left( \frac{\partial \mathcal{L}_j}{\partial \lambda_m} \right) \right|_{s \neq m} &= \beta_m Y_{ms} V_m \cos(\theta_{ms} + \delta_s - \delta_m) \\ &\quad + Y_{ms} V_m \sin(\theta_{ms} + \delta_s - \delta_s) \end{aligned} \quad \text{A.50}$$

$$\left. \frac{\partial}{\partial V_s} \left( \frac{\partial \mathcal{L}_j}{\partial \lambda_m} \right) \right|_{s=m} = \beta_s \left( \frac{P_s}{V_s} + Y_{ss} V_s \cos(\theta_{ss}) \right) - \frac{Q_s}{V_s} + Y_{ss} V_s \sin(\theta_{ss}) \quad \text{A.51}$$

$$\frac{\partial}{\partial \lambda_s} \left( \frac{\partial \mathcal{L}_j}{\partial \lambda_m} \right) = 0 \quad \text{A.52}$$

$$\frac{\partial}{\partial \lambda_{sj}} \left( \frac{\partial \mathcal{L}_j}{\partial \lambda_m} \right) = 0 \quad \text{A.53}$$

$$\begin{aligned} \frac{\partial}{\partial \delta_s} \left( \frac{\partial \mathcal{L}_j}{\partial \lambda_{mj}} \right) \Big|_{s \neq m \neq j} &= -\alpha_{mj} Y_{js} V_j V_s \sin(\theta_{js} + \delta_s - \delta_j) \\ &+ Y_{ms} V_m V_s \sin(\theta_{ms} + \delta_s - \delta_m) \end{aligned} \quad \text{A.54}$$

$$\frac{\partial}{\partial \delta_s} \left( \frac{\partial \mathcal{L}_j}{\partial \lambda_{mj}} \right) \Big|_{s=m \neq j} = -\alpha_{sj} Y_{js} V_j V_s \sin(\theta_{js} + \delta_s - \delta_j) + Q_s + Y_{ss} V_s^2 \sin(\theta_{ss}) \quad \text{A.55}$$

$$\begin{aligned} \frac{\partial}{\partial \delta_s} \left( \frac{\partial \mathcal{L}_j}{\partial \lambda_{mj}} \right) \Big|_{s=j \neq m} &= \alpha_{mj} (-Q_j - Y_{jj} V_j^2 \sin(\theta_{jj})) \\ &+ Y_{mj} V_m V_j \sin(\theta_{mj} + \delta_j - \delta_m) \end{aligned} \quad \text{A.56}$$

$$\begin{aligned} \frac{\partial}{\partial V_s} \left( \frac{\partial \mathcal{L}_j}{\partial \lambda_{mj}} \right) \Big|_{s \neq m \neq j} &= \alpha_{mj} Y_{js} V_j \cos(\theta_{js} + \delta_s - \delta_j) \\ &- Y_{ms} V_m \cos(\theta_{ms} + \delta_s - \delta_m) \end{aligned} \quad \text{A.57}$$

$$\frac{\partial}{\partial V_s} \left( \frac{\partial \mathcal{L}_j}{\partial \lambda_{mj}} \right) \Big|_{s=m \neq j} = \alpha_{sj} Y_{js} V_j \cos(\theta_{js} + \delta_s - \delta_j) - \frac{P_s}{V_s} - Y_{ss} V_s \cos(\theta_{ss}) \quad \text{A.58}$$

$$\frac{\partial}{\partial V_s} \left( \frac{\partial \mathcal{L}_j}{\partial \lambda_{mj}} \right) \Big|_{s=j \neq m} = \alpha_{mj} \left( \frac{P_j}{V_j} + Y_{jj} V_j \cos(\theta_{jj}) \right) \quad \text{A.59}$$

$$-Y_{mj} V_m \cos(\theta_{mj} + \delta_j - \delta_m)$$

$$\frac{\partial}{\partial \lambda_s} \left( \frac{\partial \mathcal{L}_j}{\partial \lambda_{mj}} \right) = 0 \quad \text{A.60}$$

$$\frac{\partial}{\partial \lambda_{sj}} \left( \frac{\partial \mathcal{L}_j}{\partial \lambda_{mj}} \right) = 0 \quad \text{A.61}$$

## VITA

Wafa Elballa was born in Omdurman, Khartoum, Sudan to the parents of Elkhier and Samiya. She is the eldest among her three brothers Husam, Mohamed and Mujtaba. Ms. Elballa attended Ahmed Basheer El-Abbady Elementary School and continued to Asmaa Abelraheem Model High School in Omdurman, Khartoum, Sudan. In 2008, she was admitted to the University of Khartoum, Khartoum, Sudan where she earned her Bachelor of Science (Honors) in Electrical Engineering with First Class. Her degree was focused on Power Systems Engineering. After graduation, Ms. Elballa worked as a part-time teaching assistant at the Department of Electrical and Electronic Engineering at the University of Khartoum. In August 2015, Ms. Elballa accepted a graduate assistantship at the University of Tennessee at Chattanooga where she started to pursue her master's degree in Electrical Engineering. Graduating in August 2017, Ms. Elballa is now considering joining a PhD program where she can continue her educational path.