Brunel
UNIVERSITY
WEST LONDON

A NOVEL Q-LIMIT GUIDED CONTINUATION POWER FLOW METHOD FOR VOLTAGE STABILITY ANALYSIS

A thesis submitted for the degree of

Doctor of Philosophy

by

Pengcheng Zhu

Brunel Institute of Power Systems
School of Engineering and Design
Brunel University UK
Sep 2008

Abstract

Voltage security assessment is becoming a more and more important issue due to the fact that electrical power systems are more prone to voltage instability under increased demand, and it can be time-consuming to determine the actual level of voltage security in large power systems. For this reason, this thesis presents a novel method for calculating the margin of voltage collapse that is based on the Continuation Power Flow (CPF) method. The method offers a flexible and reliable solution procedure without suffering from divergence problems even when near the bifurcation point. In addition, the new method accounts for reactive power limits. The algorithmic continuation steps are guided by the prediction of Q-limit breaking point. A Lagrange polynomial interpolation formula is used in this method in order to find the Q-limit breaking point indices that determine when the reactive power output of a generator has reached its limit. The algorithmic continuation steps will then be guided to the closest Q-limit breaking point, consequently reducing the number of continuation steps and saving computational time. The novel method is compared with alternative conventional and enhanced CPF methods. In order to improve CPF further, studies comparing the performance of using direct and iterative solvers in a power flow calculation have also been performed. I first attempt to employ the column approximate minimum degree (AMD) ordering scheme to reset the permutation of the coefficient matrix, which decreases the number of iterations required by iterative solvers. Finally, the novel method has been applied to a range of power system case studies including a 953 bus national grid transmission case study. The results are discussed in detail and compared against exiting CPF methods.

Acknowledgements

I would first like to express my cordially gratitude to my supervisor Dr. Gareth Taylor. I especially appreciate tremendous effort, support, and unwavering patience which he gave me through out my study. I also thank him for helping, encouragement, and supervision in these years when I study in Brunel University. I truly believe that the level of my research skills and knowledge has been significantly improved simply because he demanded, expected nothing less. His accomplishments in power system domain, visionary thoughts, energetic working style and unceasing exploring spirit in academe will influence all my life. Besides being a great teacher, the advice I've received has been beneficial in my whole life.

I also like to thank Prof. Malcolm Irving as my second supervisor for his valuable time in evaluating this thesis. My sincerely thanks is also given to my research colleagues at Brunel Institute of Power Systems (BIPS) and all staff of BIPS.

Last, but not least, to my parents, my sister and whole extended family. Thanks for their constant support, encouragement, and understanding whenever I am facing difficulties or being in troubles during the years of my study, especially to my parents who sponsor me all the expenses of my study in UK and my girlfriend who is the source of my confidence to finish this thesis.

Contents

List of Figures

List of Tables

Chapter 1: Introduction

This chapter introduces the technical background of my research work, the developmental history of continuation power flow and the achievements that numerous researchers have made to improve the performance so far. It is clearing emphasized that a more efficient voltage stability analysis method is required in order to meet the development of modern power systems. The contribution and structure of this thesis is also presented in this chapter.

1.1 Introduction and Background

As we know that the power systems are becoming more complex and more heavily loaded in recent years, the transmission networks are becoming larger and larger, which result in the power systems needing improved technologies and algorithm procedures to support their efficient operation. The degree of operational uncertainty has increased with the proliferation of more deregulated and competitive power systems. Such power systems are more prone to voltage instability under increased demand and it can be time-consuming to determine the actual level of voltage security in large power systems [1, 2, 3, 4, 5, 6]. Voltage security assessment is becoming a more and more important issue in electrical power systems [7, 8], it is often necessary for operators and planners to quickly analyze voltage stability margins for a vast number of network scenarios in order to effectively secure a power system against voltage collapse [9].

Continuation Power Flow (CPF) [10, 11] is an established method that analyses voltage stability by tracing the P-V curve using a predictor and corrector scheme [12, 13], in this way the method remains well-conditioned at and around the bifurcation point. CPF methods are popular techniques used in the voltage stability analysis of power systems all over the world. Due to the rapid development of modern power system economics, it is necessary to improve or find a more efficient voltage analysis method to meet the requirements of present and future power system security. Against this background, I have investigated, developed and demonstrated a novel Q-limit guided continuation power flow (GCPF) method that is based on the standard CPF. The method has also been applied to realistic large-scale network problems. In additional work, I explored the performance of using different linear solvers in a power flow computation.

1.1.1 Significance of CPF Computation

In recent years, modern economic and environmental pressures have resulted in continuing interconnection of bulk power systems and the increasing utilization of existing facilities. This has led to increasingly more complex and larger power systems that are being pressed to operate at levels ever closer to the system limits. This operating condition has contributed to the growing importance of problems associated with voltage stability analysis of power systems. Power transmission capability has traditionally been limited by either rotor angle stability or by thermal loading capabilities. However, over the last two decades voltage stability has emerged as a major concern in both of the planning and operating of power systems [3, 8].

Voltage collapse is one of the problems caused by voltage instability. Generally two types of system disturbances would lead to voltage collapse: load variations and contingencies. Several recent power system blackouts were directly related to voltage collapse, which is characterized by a slow variation in the system operating point in such a way that voltage magnitudes at load buses gradually decrease until a sharp, accelerated change occurs. Voltage collapse has been especially experienced by heavily loaded power systems subject to an increase in load demands. There has been a wide consensus that as power system operates under increasingly stressed conditions, the ability to maintain voltage stability in order to avoid collapse becomes a serious concern [1].

An increasing number of electric utilities are facing voltage stability-imposed limits. voltage instability or collapse has resulted in several major system failures. Case in point are the occurrence in Sweden in 1982 caused by a contingency, the New York blackout in July 1977, power failures across France in December 1978, January 1987 and similar experiences in other countries world wide [3].

On August 14, 2003, large portions of the Midwest and Northeast United States and Ontario, Canada, experienced an electric power blackout. The outage affected an area with an estimated 50 million people and 61,800 megawatts (MW) of electric load in the states of Ohio, Michigan, Pennsylvania, New York, Vermont, Massachusetts, Connecticut, New Jersey and the Canadian province of Ontario. The Blackout was caused by deficiencies in specific practices, equipment, and human decisions by various organizations that affected conditions and outcomes [94].

The power system failure occurred on 23 July 1987 in the service area of the Tokyo Electric Power Company. The unusually hot weather led to concentrated mass consumption of electric power, causing the blackout which affected 2.8 million customers in the area. There are three considerable causes of this failure [2]

- the power demand level on that day was unusually high,
- the speed of the demand increase exceeded the level of previous experiences,
- Air conditioners which have the characteristics of voltage down causing current rise are in wide-spread use in Japan.

The supply of electric power is becoming more and more scantily sometime. In order to have a reliable and economical electric power supply, the power system is prone to become larger. The distance of electrical power transmission is becoming longer. So the calculation of the margin of voltage collapse is very important for the power system security.

There is a very strong economical motivation to improve the accuracy and the speed of voltage collapse point computation in competitive power systems today. Voltage collapse can occur in a power system following a progressive decline at the bifurcation point, usually due to load increasing. Lack of adequate reactive power resources in a power system has been recognized as a major contributing factor in a voltage collapse process. As we know, once a reactive power resource has reached its limit, it can no longer adjust the voltage. Consequently, large load increasing may result in promote voltage decline. Although there are some other factors that influence the voltage collapse process, we shall only focus on this major factor of reactive power resources reaching their Q-limit.

In order to avoid voltage collapse, we have to control the voltage of power systems within the security margin. That means if we can calculate the maximal demand of the load that the power system can afford without encountering collapse, then we can operate the power system safely. Due to this purpose, some methods to calculate the voltage collapse point (fold point, saddle-nose point, bifurcation point and critical point) have been created. CPF is one of the most popular and reliable methods used around the world.

1.1.2 Voltage Collapse of Power System

As power systems become more complex and more heavily loaded, voltage collapse becomes an increasingly serious problem. Voltage collapse has already occurred in realistic power systems.

A power system is an electrical network containing components such as generators, transmission lines, loads, and voltage controllers. Practical networks are large, ranging from hundreds to thousands of buses and branches. Since the basic practical functions of a power system are the generation and distribution of electric power, generators are essential components. Under normal operating conditions, a generator is essentially a constant-voltage source. But in a transient condition, the excitation and rotor dynamics can produce undesirable oscillations of the system frequency and voltage magnitude.

As for the components of the system, a transmission line can be modelled by a series RL branch with shunt capacitors. The control system is critical, and has benefited from recent technological advances. The turns-ratios of some transformers, for example, are automatically adjusted by onload tap-changers to maintain the voltage levels near the loads. Although control mechanisms have increased in number and sophistication, the networks themselves are very complex and can behave in ways that are difficult to predict.

Load devices themselves vary greatly, from resistive lighting devices to dynamic components such as large induction motors. And power systems can be interconnected to allow exchange of electric power between different utility systems. A large, nonlinear, interconnected power network can exhibit very complex dynamic phenomena when the system is disturbed from a steady-state operating condition.

To complicate things even more, the power systems are becoming more heavily loaded as the demand for electric power rises, while economic and environmental concerns constrain the construction of new transmission and generation capacity. Under these stressful operating conditions, we are encountering a new instability problem called voltage collapse, which has led to blackouts in electric utilities around the world [3].

1.1.3 Voltage Stability Analysis Methods

Voltage security assessment is becoming a more and more important issue in electric power systems [7, 8], it is often necessary for operators and planners to quickly analyze voltage stability margins for a large number of network scenarios in order to effectively secure a power system against voltage collapse [9]. For this reason, various methods have been developed for voltage stability analysis, such as the repetitive method, which repeatedly computes power flows with increasing power load demand in order to reach the critical point. However, the power flow solution is prone to divergence due to the Jacobian matrix becoming singular near the saddle-node bifurcation point [11]. Continuation Power Flow (CPF) [10, 11] is an established method that solves the problem by tracing the P-V curve using a predictor and corrector scheme [12, 13], in this way the method remains well-conditioned at and around the bifurcation point.

In order to speed up the CPF computation, numerous researchers have investigated new voltage collapse index calculation methods [14-28] and how to improve the performance of conventional CPF [29, 30, 31, 32]. A considerable number have made use of a nonlinear predictor instead of a conventional linear tangent or secant predictor [29], the linear predictor employs a small step-size in the CPF computing process, but the nonlinear predictor is not restricted to a small step-size. In other words, the nonlinear predictor enables the CPF method to take a larger step-size than the linear case, which increases the computational speed of the CPF methods. Other researchers dynamically adapt the feasible step-size in order to speed up the CPF computation [30]. A large step-size is used in the 'flat' part of the solution curve and a small step-size is used in the part with a high degree of curvature. Most of the approaches mentioned above are reasonably efficient, but it is still necessary to improve the performance of CPF methods further in order to meet the requirements of present and future operational time-scales.

1.1.4 Application of Linear Solvers in Power Flow Solution

The power flow is usually solved by the Newton-Paphson solution method. The main step is the linearization of the nonlinear power flow equations and the subsequent solution of this linear system. All Newton-Raphson based power flow algorithms have in common that one large or two smaller sized linear system of equations must be solved during each Newton-Raphson iterative step. So, to find an appropriate linear solver for these linear equations becomes an interesting project, numerous researchers have investigated the performance of different linear solvers used in power load computation.

Besides the well known direct Gaussian elimination method, the application of iterative linear solvers in power flow computation has already appeared in recent years. For example, the first application of the Conjugate Gradients (CG) method [33] to the decouple power flow has been described in [34, 35]. For the fast decoupled power flow both papers state a significant performance improvement of CG based methods comparing to a direct solution. From all power flow approaches known only the decoupled power flow satisfies the CG conditions of positive definite and symmetric linear system matrices. The first application of non-stationary, iterative methods to the Newton-Raphson power flow approach has been described in [36]. The main distinction from the CG methods lies in the fact that the iterative solvers investigated in [36] is also applicable to asymmetric and indefinite linear system matrices. With the development of mathematical algorithms, there are many iterative linear solvers known today can solve asymmetric, indefinite and sparse linear system matrices. Practical usage of these iterative linear solvers is only reached when preconditioning is applied to the linear system of equations, good preconditioning will group the eigenvalues of the transformed linear system matrix together and will thus result in faster convergence [37, 38].

It is well known that the CPF is a continuous process of power flow solutions starting at a base load and leading to the steady state voltage stability (critical point) of the system, on the other hand, solving linear equations of the form Ax = b plays very important role and takes the most computational time in Newton-Raphson power flow methods [39], so an efficient linear solver is one of the factors in speeding up CPF computation.

At present, the direct method (Gaussian elimination) is used commonly in solving large sparse sets of linear equations [40-44] in Newton-Raphson solution methods, but the Gaussian elimination method is hard to be parallelized or factorized, which restricts the development and improvement of direct methods. The iterative methods for solving large sparse sets of linear equations can naturally profit from parallelization and factorization [39, 45, 46], and a good preconditioning will effect the eigenvalues of the transformed linear system matrix together and will thus result in faster convergence [37].

In Newton-Raphson power flow methods, the Jacobian coefficient matrix is an unsymmetrical, possibly indefinite matrix [37, 47], according to this characteristics, there are a number of iterative methods can be employed. We may consider the generalized minimum residual method (GMRES) [48], bi-conjugate gradient method (BiCG) [49] and bi-conjugate gradient stabilized method (BiCGStab) [50]. Comparative studies between direct and iterative methods have been implemented and discussed in this thesis.

1.2 Contribution of the thesis

This thesis concentrates on the improvement of performance of voltage stability analysis methods. Firstly, we proposes a novel Q-limit guided continuation power flow (GCPF) method [51], the method accounts for reactive power limits. The GCPF method is developed in order to improve the performance of the standard CPF method in the aspects of robustness and computational efficiency. The GCPF method improves the standard CPF, but is based on the standard CPF method, which offers a flexible and reliable solution procedure without suffering from divergence problems even near the bifurcation point. This part of work has been published in IEEE PES 2008 general meeting proceedings on July 2008 in USA.

Traditionally, voltage stability limits were considered only in terms of the saddle node bifurcation point [11]. However, there exists another type of bifurcation point that causes instability [52], this type of instability may occur when a state variable reaches a limit [53]. In practice this is especially significant with regard to generator reactive power limits (Q-limits). In fact, in power flow based voltage stability assessment, the Q-limit instability frequently occurs at a point at which the reactive power output of voltage targeted generator reaches a Q-limit [12, 54].

I propose a more efficient method, which is referred to as a Q-limit GCPF [51] method. It employs an original step-size control rule that is fundamentally different from the conventional feasible step-size as controlled by the gradient of solution curve [30]. The step-size is controlled by calculating the distance of continuous Q-limit breaking points and we make use of a nonlinear Lagrange polynomial interpolation technique [55] in order to predict at which point the reactive power limit of a generator is reached. The closest Q-limit breaking point is then taken as the next continuation step. Therefore, the continuation steps are efficiently guided by the prediction of Q-limit points and hence the step size is controlled in a more adaptable manner than the conventional rule based method [30], the GCPF method significantly reduces the number of algorithmic continuation steps. A comparison of computing time between

GCPF, conventional CPF (CCPF) [9, 11] and Enhanced CPF (ECPF) [29] that using nonlinear predictor method has been performed.

Secondly, we have investigated the application of different linear solvers in the Newton-Raphson power flow computation in terms of computational time and accuracy. It is well known that efficient solvers for linear equations on the form Ax = b are essential and most of the computation time is spent in solving the linear equations in the power flow calculation [39]. The iterative methods for solving large sparse sets of linear equations can naturally profit from parallelization and vectorization [39], and direct methods, such as Gaussian elimination, are difficult to parallelize or factorize. Comparison studies between direct and some iterative methods have been performed. This part of work has been submitted for review to IET Generation, Transmission & Distribution on May 2008.

Solving a linear equation of the form Ax = b plays an important role in power flow calculation, and uses the most computational time in a power flow. CPF is a continuous process that computes many power flow solutions, so an efficient linear solver is necessary to speed up a CPF method. In this thesis, studies compare the performance of direct and iterative solvers. It is well established that iterative methods for solving large sparse sets of linear equations can naturally profit from parallelization and factorization.

The direct method investigated in this research is the Gaussian Elimination method, which solves the linear equations in a finite number of steps and may require extensive computational storage and arithmetic processing. Gaussian Elimination method has been improved over the years through the use of efficient ordering techniques and sparse programming. However, the computation time of a direct method can still be the limiting factor in cases involving very large systems of equations or numerous repetitive solutions [34]. However, the direct method is the most commonly used method in power systems analysis.

In the Newton-Raphson power flow, the Jacobian coefficient matrix is an asymmetric, possibly indefinite matrix [47]. For asymmetric linear systems, there are a number of iterative methods that can be employed. We may consider the generalized minimum residual method (GMRES), bi-conjugate gradient method (BiCG) and bi-conjugate gradient stabilized method (BiCGStab). Comparative studies between direct and iterative solvers have been implemented and discussed in this thesis.

It has been reported that the use of some iterative methods for power flow solutions decreases the computation time, relative to the LDU factorization based direct methods, for large power system problems [34, 35]. In these studies, the incomplete LU factorization (ILU) of the Jacobian matrix is used to accelerate the GMRES, BiCG and BiCGStab methods as the preconditioner, and we first attempt to employ the column approximate minimum degree (AMD) ordering scheme [56] to reset the permutation of the Jacobian coefficient matrix, which are quite effective in clustering the eigenvalues of the linear system [37, 47].

Finally, we create a package of m-files for solving CPF problems in Matpower [57], it is intended as a simulation tool for researchers and educators that are easy to use and modify. The code is designed to give the best performance possible while keeping the code simple to understand and modify.

1.3 Publications resulting from this thesis

The following publications have been derived form this thesis.

- Pengcheng Zhu, Gareth Taylor and Malcolm Irving, "A Novel Q-limit Guided Continuation Power Flow Method," *IEEE PES General Meeting 08 Proceeding*, Pittsburgh USA, July 2008. (Published)

- Pengcheng Zhu, Gareth Taylor and Malcolm Irving, "Performance Analysis of a Novel Q-limit Guided Continuation Power Flow Method", *IET Generation, Transmission & Distribution*, Submitted for review on May 2008.

1.4 Structure of the thesis

This thesis is organized as follows:

Chapter 1 introduces the technical background of my research, the significance of voltage stability analysis methods and the achievements that numerous authors have made on the improvement of performance of CPF methods so far. The contribution of my research work is also presented in this chapter.

Chapter 2 provides a technical description of the conventional continuation power flow (CCPF), which is a classic voltage stability analysis tool and has been widely used all over the world.

Chapter 3 introduces some variation work of standard CPF. One is the enhanced CPF (ECPF) method, which employs a nonlinear predictor to control the step size; the other one is my proposed GCPF method, details of GCPF algorithm procedures are described in this chapter.

Chapter 4 introduces the developmental history of the project of applications of iterative solvers in power flow calculation, and also presents the summary of AMD ordering scheme, ILU factorization method and linear solver approaches, which have been investigated in my research work.

Chapter 5 introduces details of modification and development of Matpower to simulate the experiments. This chapter includes the mathematical model of GCPF method. I also modify the code to implement Newton-Raphson power flow using different iterative linear solvers. I create a tool package of m-files for solving CPF solutions based on Matpower.

Chapter 6 is the full results of comparison between CCPF, ECPF and our proposed GCPF method. I have done the comparisons in multiple means, which include illustration figures, numerical results and discussions. A performance comparison is made between CCPF, ECPF and GCPF methods in terms of required continuation steps, computational time and accuracy. These CPF methods have been applied to a large range of power system case studies including two National Grid (NG) transmission case studies.

Chapter 7 is the results of comparison and analysis between direct and iterative linear solvers employed in a power flow calculation. The effect of AMD ordering scheme based ILU factorization preconditioning on eigenvalue distribution, convergence rate and computing time is illustrated and discussed in multiple means. The comparison work has been applied to a large range of case studies, such as Poland 2736 bus system.

Chapter 8 summarizes the entire thesis conclusions, including two parts of conclusions. One part is from the comparison of performance between CCPF, ECPF and GCPF methods; another part is form the comparison of using different linear solvers in the power flow calculation. On the other hand, possible future works are suggested and discussed in this chapter.

Chapter 2: Conventional Continuation Power Flow Method

This chapter presents a technical description of conventional continuation power flow (CCPF), which is a classic voltage stability analysis method and has been widely used all over the world. This method adapts predictor and corrector scheme to trace the P-V curve, in this way the method remains well-conditioned at and around the bifurcation point.

2.1 Introduction

Continuation method [26, 58] is a mathematical method, which has been applied successfully to a variety of engineering problems including electric power systems [59, 60]. CPF is a method based on the continuation method, which can trace the power flow solution curve with respect to the varying parameters [10, 11, 61-65]. The paper [10, 11] was the first published literatures to introduce the application of CPF method to power systems. The CPF can pass through the saddle-node bifurcation point without any numerical difficulty. CPF uses EPRIIPFLOW [66] as a platform.

CPF becomes a comprehensive tool for tracing power system steady-state stationary behaviour due to parameter variations, sometimes called curve tracing or path following, are useful tools to generate solution curves for general nonlinear algebraic equations with a varying parameter.

The main advantages of CPF over repetitive power flow calculations are:

- It is more reliable than the repeated power flow approach in obtaining the solution curve; especially for ill-conditioned power flow equations.
- It is faster than the repeated power flow approach via an effective predictor-corrector, adaptive step-size selection algorithm.

The CPF method can be used in a variety of applications such as: to analyze voltage problems due to load and/or generation variations, to calculate maximum interchange capability of power between two areas and maximum transmission capability, to simulate power system static behaviour due to load and/or generation variations.

2.2 System Model

Consider a comprehensive (static) power system model expressed in the following form [67, 68]:

$$f(x,\lambda) = F(x) + \lambda b = 0 \tag{2-1}$$

where $x \in R^n$ is the state vector representing the bus voltage magnitudes and angles, $\lambda \in R^1$ is a (controlling) parameter subject to variation and $b \in R^n$ represents the change in real and reactive power load demand and the change in real power generation [54, 69]. Using terminology from the field of nonlinear dynamical systems, system (2-1) is a one-parameter nonlinear system. In power system applications, a one-parameter dynamical system is a system together with one of the following conditions:

1. The reactive (or real) power demand at one load bus varies: and the real power generations at some collection of generator buses vary, and their variations can be parameterized while the others remain fixed.
2. Both the real and reactive power demand at a load bus vary: and the real power generation at some collection of generator buses vary, and their variations can be parameterized, again the others remain fixed.
3. The real and/or reactive power demand at some collection or load buses varies; and the real power generations at some collection of generator buses vary, and their variations can be parameterized while the others are fixed.

Generally speaking, power systems are dynamical systems and are normally operated near a stable equilibrium point. As system loads and generations change slowly, the stable equilibrium point changes position but remains as a stable equilibrium point. This situation may be modelled with the static model (2-1) by regarding $f(x,\lambda)=0$ as specifying the position of the stable equilibrium point, x as a function of λ. This model may also be called a parametric power flow model.

For example, if the system represented is initially near a stable equilibrium point $x_s(\lambda)$, then the dynamics will make the system state track $x_s(\lambda)$ as λ slowly varies.

Exceptionally, variations in λ will cause the stable equilibrium point to bifurcate. The stable equilibrium point may then disappear or become unstable depending on the way in which the parameter is varied and on the specific structure of the system. One typical way in which system (2-1) may lose stability is that the stable equilibrium points $x_s(\lambda)$ and an unstable equilibrium point $x_1(\lambda)$ coalesce and disappear in a saddle-node bifurcation as parameter λ varies. The nose point of PV and QV curve is an example of the saddle-node bifurcation commonly used in the power industry to analyse voltage stability and voltage collapse.

We next discuss an indirect method to simulate the approximate behaviour of the power system (2-1) due to load and/or generation variation. Before reaching the critical point, the power system with a slowly varying parameter traces its operating point which is a solution of the equation (2-1) whose corresponding Jacobian has all eigenvalues with negative real parts.

The indirect method is to start from the operating point of the power system (x_0, λ_0), and produce a series of solution points (x_i, λ_i) in a prescribed direction, determined by participating load and generation variations, until the critical point is reached.

It is known that the set of power flow equations (2-1) near the critical point is ill-conditioned as Jacobian matrix becomes singular, making the Newton method diverge in the neighbourhood of the critical point. From a numerical analysis viewpoint, this is due to the fact, that at the critical point the two equilibrium points coalesce to form an equilibrium point x^*. The Jacobian matrix evaluated at this point x^* has one zero eigenvalue, causing the set of power flow equations to be ill-conditioned.

There are several possible means to resolve the numerical difficulty from the ill-conditioning. One effective way is to introduce the arc-length s on the solution curve as a new parameter in continuation process. The step size along the arc-length s has the following constraint:

$$\sum_{i=1}^{n}(x_i - x_i(s))^2 + (\lambda - \lambda(s))^2 = \Delta s^2 \qquad (2\text{-}2)$$

Where s is the arc-length and Δs is the step size.

At this stage we can introduce a geometrical representation of the arc length s as illustrated in figure 2-1. As step size tends to zero, the chord Δs tends to equal s, which allows us to then introduce the parameter Δs, that is the chord subtended by the arc-length s.

So, the simultaneous equations (2-1) and (2-2) is well conditioned, even at the critical point. This method solves the simultaneous equations to obtain the solution curve passing through the critical point without encountering the numerical difficulty of ill conditioning.

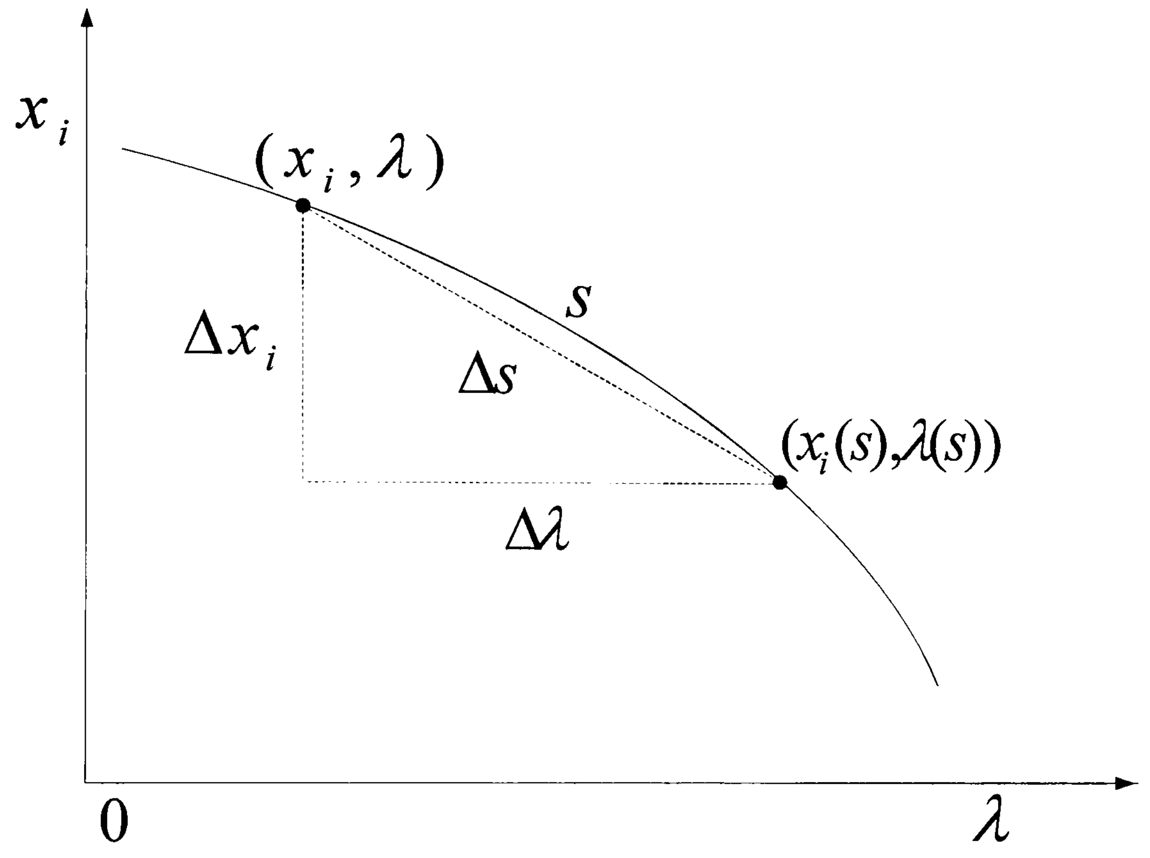


Figure 2-1: Geometrical Representation of CPF Variables

Continuation methods, sometimes called curve tracing or path following, are useful tools to generate solution curves for general nonlinear algebraic equations with a varying parameter. CPF uses continuation methods to trace power system steady state behaviour due to load and generation variation. Continuation method has four basic elements:

- Parameterization
- Predictor
- Corrector
- Step-size control

2.2.1 Parameterization

Parameterization is a mathematical way of identifying each solution on the solution curve so that 'next' solution or 'previous' solution can be quantified. There are three different types of parameterisations:

1. Physical parameterization using the controlling parameter λ, in which case the step length is $\Delta\lambda$.
2. Local parameterization, which uses either the controlling parameter λ or any component of the state vector x; namely x_k to parameterize the solution curve. The step length in the local parameterization is $\Delta\lambda$ or Δx_k.
3. Arclength parameterization employing the arclength along the solution curve to perform parameterisation, the step length in this case is Δs:

$$\Delta s = \sum_{i=1}^{n} \left\{ (x_i - x_i(s))^2 + (\lambda - \lambda(s))^2 \right\}^{0.5} \qquad (2\text{-}3)$$

the arclength parameterization can use different weighting factors (instead of an equal weighting factor) in the above equation.

While using the controlling parameter to parameterize the solution curve has physical significance, it encounters numerical difficulties in the vicinity of critical point. In order to resolve this problem and to design an effective predictor, CPF makes use of the arc-length parameterization.

2.2.2 Predictor

The purpose of the predictor is to find an approximation point for the next solution. Suppose we are at the $i-th$ step of the continuation process and the $i-th$ solution $\left(x^i, \lambda^i\right)$ of (2-1) has been found. The predictor attempts to find an approximation point for the next solution (x^{i+1}, λ^{i+1}). The quality of the approximation point by a predictor significantly affects the number of iterations required by a corrector in order to obtain an actual solution. A better approximation point yields a fewer number of iterations needed by a corrector to reach the solution. Several different predictors have been introduced in the literature of numerical analysis. They can be divided into two classes:

1. ODE based methods, which use the current solution and its derivatives to predict the next solution. The tangent method, a popular one as a predictor, is a first order ODE-based method;
2. Polynomial extrapolation based methods, which use only current and previous solutions to find an approximated solution. The secant method, a popular polynomial-based predictor, uses the current solution and the previous one to predict the next one.

2.2.2.1 Tangent Method

The tangent method calls for the calculation of the derivatives of $x_1, x_2, \cdots, x_n, x_{n+1}$ with respect to the arclength s:

$$\frac{dx_1}{ds}, \dots, \frac{dx_n}{ds}, \frac{dx_{n+1}}{ds} \tag{2-4}$$

To find these derivatives, differentiate both sides of equation (2-1) with respect to s:

$$f_{x_1}\frac{dx_1}{ds}+\cdots+f_{x_{n+1}}\frac{dx_{n+1}}{ds}=0 \tag{2-5}$$

Equation (2-5) is an implicit system of n linear algebraic equations in $n+1$ unknowns

$$\frac{dx_i}{ds}, i=1,\cdots,n+1 \tag{2-6}$$

with the coefficients being the elements of the matrix:

$$Df=\begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_k} & \cdots & \frac{\partial f_1}{\partial x_{n+1}} \\ \frac{\partial f_2}{\partial x_1} & \cdots & \cdots & \cdots & \cdots & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_k} & \cdots & \frac{\partial f_n}{\partial x_{n+1}} \end{bmatrix} \tag{2-7}$$

the following equation is required to make sure that s is the arc-length on the curve.

$$(\frac{dx_1}{ds})^2+\cdots+(\frac{dx_{n+1}}{ds})^2=1 \tag{2-8}$$

note that equations (2-5) and (2-8) form a set of $n+1$ equations in $n+1$ variables. Also notice that (2-5) is a set of linear equations in its $n+1$ unknowns and (2-8) is nonlinear. A special method to solve this $n+1$ equations according to Kubicek [60] is as follows:

Suppose

$$dx_k/ds \neq 0 \tag{2-9}$$

for some k, $1 \le k \le n+1$, and let Df_k be the matrix of Df with the k-th column taken out, and suppose Df_k is not singular, then equation (2-5) can be solved for the unknows

$$\frac{dx_1}{ds},\cdots\frac{dx_{k-1}}{ds},\frac{dx_{k+1}}{ds},\cdots\frac{dx_{n+1}}{ds} \tag{2-10}$$

in the form of

$$\frac{dx_i}{ds} = \beta_i \frac{dx_k}{ds} \qquad i \neq k, i = 1, \cdots, n+1 \tag{2-11}$$

the solution for the coefficients β_i can be obtained by applying Gaussian elimination to the matrix Df_k. In fact, if one performs the Gaussian elimination with pivoting to the full set matrix Df, the column index k is also found.

Substituting (2-11) to (2-8), one gets

$$(\frac{dx_k}{ds})^2 = (1 + \sum_{i=1, i\neq k}^{n+1} \beta_i^2)^{-1} \tag{2-12}$$

Equations (2-11) and (2-12) constitute the explicit expression of the derivatives of the curve $x(\lambda)$ passing through the j-th continuation point (x^j, λ^j) with respect to arc-length s. A predictor step can be accomplished by integrating one step further in the prescribed direction with the step size h:

$$\hat{x}_j^{j+1} = x_j^{j+1} + h\frac{dx_j}{ds}, j = 1, \cdots, n+1 \tag{2-13}$$

In the context of computational efficiency, one has to keep in mind that the evaluation of the β_i's involves solving a set of linear algebraic equations which could be time-consuming. Thus it is advantageous to use numerical procedures which require fewer such evaluations. This consideration prompts the use of the secant method as a predictor after the tangent method produces two approximate points.

2.2.2.2 Secant Method

The polynomial extrapolation methods are based on a polynomial of varying order that passes through the current solution and previous solution (x^i, λ^i), (x^{i-1}, λ^{i-1})... to provide an approximation point for the next solution (x^{i+1}, λ^{i+1}). A trivial predictor is the zero-order polynomial which uses the current solution as an approximation point for the next solution.

$$(\hat{x}^{i+1}, \hat{\lambda}^{i+1}) = (x^{i}, \lambda^{i}) \tag{2-14}$$

A slightly modified predictor based on the zero-order polynomial is:

$$(\hat{x}^{i+1}, \hat{\lambda}^{i+1}) = (x^{i}, \lambda^{i+1}) \tag{2-15}$$

A predictor, known as the secant predictor, uses a first-order polynomial (a straight line) passing through the current and previous solutions to predict the next solution.

$$(\hat{x}^{i+1}, \hat{\lambda}^{i+1}) = (x^{i}, \lambda^{i}) + h_i(x^{i} - x^{i-1}, \lambda^{i}, \lambda^{i-1}) \tag{2-16}$$

Where h_i is an appropriate step-size, predictors based on higher-order polynomial can be similarly derived. It has been experienced that lower-order predictors are more effective in practice.

In general, ($\hat{x}, \hat{\lambda}$) is not a solution of $f(x, \lambda) = 0$, rather it is an initial guess for the corrector iteration that will hopefully converge to a solution within the specified tolerance, the distance between (x^{i}, λ^{i}) and (x^{i+1}, λ^{i+1}) is called the step length. On the other hand, the measure of distance between (x^{i}, λ^{i}) and (x^{i+1}, λ^{i+1}) is given by the parameterization strategy, for example, arc-length.

The tangent and secant predictor method is illustrated in figure 2-2.

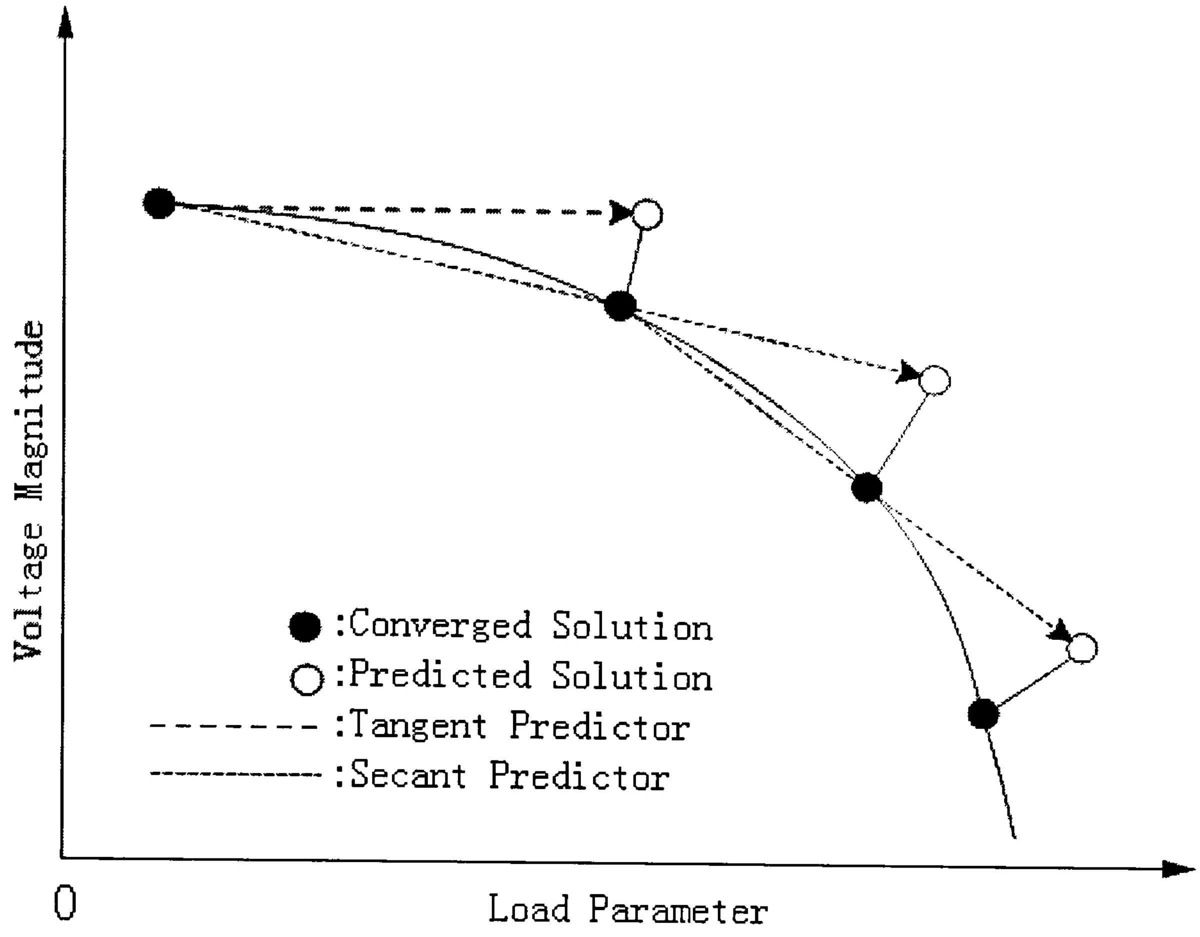


Figure 2-2: Tangent and Secant Methods

2.2.3 Corrector

After the predictor has produced an approximation $(\bar{x}^{j+1}, \bar{\lambda}^{j+1})$ for the next solution (x^{j+1}, λ^{j+1}), the error must be corrected before it accumulates. In principle, any effective numerical procedure for solving a set of nonlinear algebraic equations can be used for a corrector. Since a good predictor gives an approximation in a neighbourhood of the next solution (x^{j+1}, λ^{j+1}), a few iterations usually suffice for an appropriate corrector to achieve the needed accuracy.

The predictor-corrector scheme plays very important role in CPF methods, which is a continuous process to find a solution path. As shown in figure 2-3. It starts from a known solution and employs one step predictor to estimate a subsequent solution corresponding to the pre-set step size, this estimation is then corrected using the Newton method [70, 71]. CPF goes to the next continuation step until the critical point has been passed.

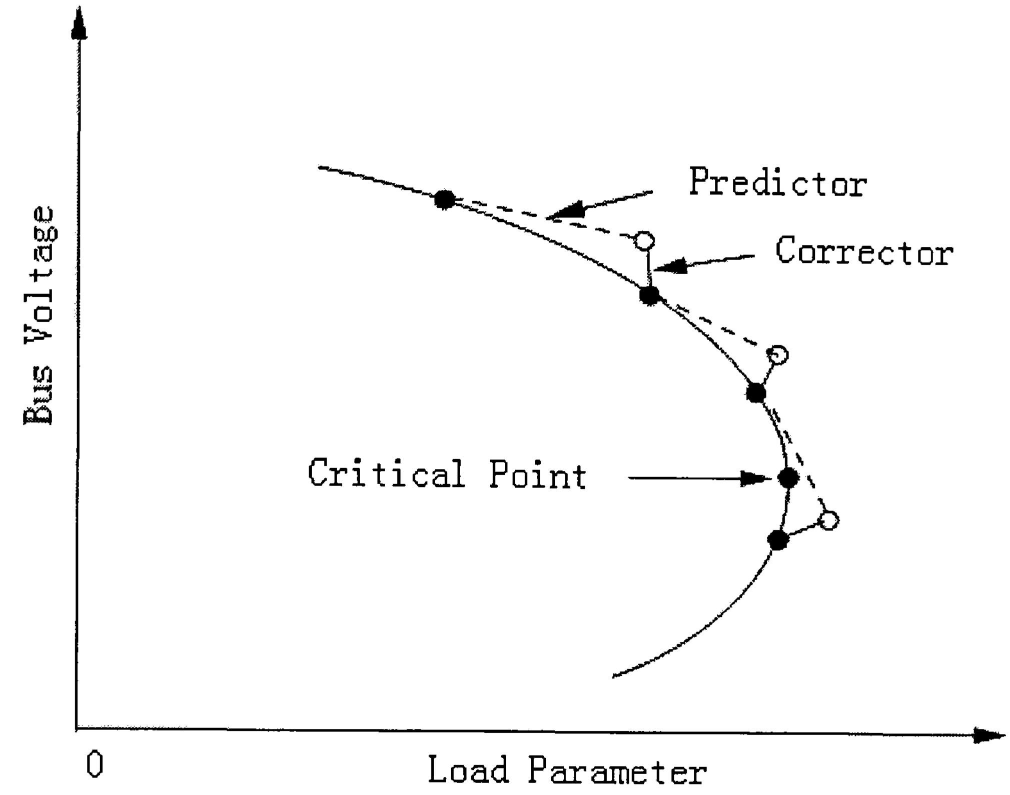


Figure 2-3: An illustration of the predictor-corrector scheme used in CPF

2.2.4 Step Size Control

One key element affecting the computational efficiency associated with a continuation method is the step-length control. It is safe to choose a constant, small step length in any continuation method. However this constant step length may often lead to inefficient computation, such as too many steps through the 'flat' part of the solution curve. Similarly, an inadequately large step length can cause the predicted point (produced by predictor) to lie far away from the (true) solution point, and as a result, the corrector needs much iteration to converge. In the extreme case, the corrector may diverge. Ideally, the step length should be adapted to the shape of the solution curve to be traced: a large step length should be used in the 'flat' part of the solution curve and a smaller step-length in the 'curly' part (part with high degree of curvature) of the solution curve. Of course, the shape of the solution curve i is unknown beforehand, making the task of designing an effective step-length control difficult. Thus good step length controls are usually custom designed for specific applications. Despite this, some general considerations may be implemented in the continuation procedure in order to improve its performance.

One strategy for step length control is to set up an upper limit $h_{\max,i}$ for each variable x_i. The actual step length h along the arclength s is the thus chosen such that:

$$h\frac{dx_i}{ds} \leq h_{\max,i}, i = 1,\cdots,n+1 \qquad (2\text{-}17)$$

the motivation for such an implementation is that the curve $x(\lambda)$ under consideration may be 'flat' with respect to some x_i, while turning sharply with respect to some other x_j. By assigning $h_{\max,i}$ accordingly, that is, giving a larger $h_{\max,i}$ to those variables along which the curve is 'flat' and smaller $h_{\max,i}$, otherwise we can make the continuation process trace quickly through the 'flat' portion of the curve and yet keep small steps through the 'curly' portion. This in turn will yield a better approximation from the predictor, thus faster convergence for the corrector. The success of this step length control method depends greatly on the proper value of $h_{\max,\ i}$ which requires prior knowledge of the problem under consideration. In the case of power system studies, experience provides good guidance. For example, the $h_{\max,\ i}$ corresponding to a bus voltage would be given a small value since the whole range for the variable is about

(0-1.2), while the $h_{\max,\ i}$ corresponding to the reactive parameter λ should be assigned a larger value.

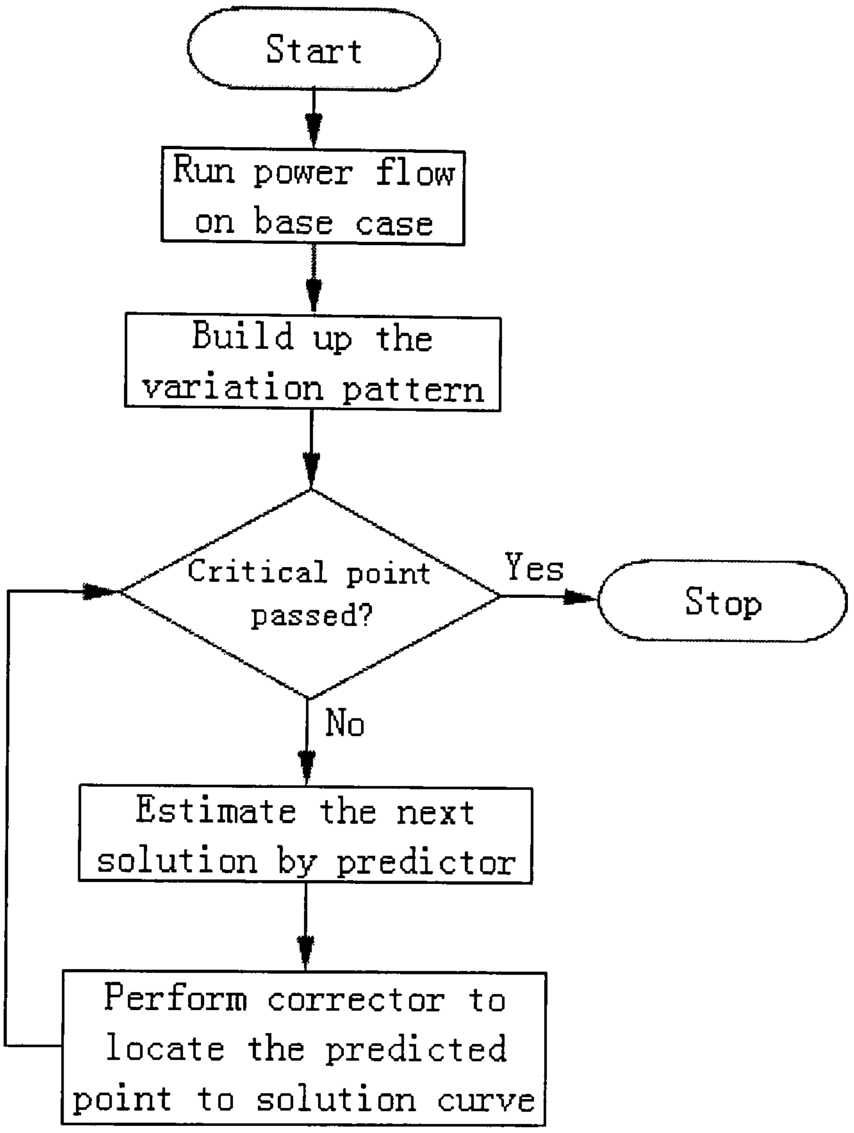

Start
Run power flow on base case
Build up the variation pattern
Critical point passed?
Yes
Stop
No
Estimate the next solution by predictor
Perform corrector to locate the predicted point to solution curve

Figure 2-4: Illustration of algorithm of predictor-corrector scheme of CPF

Another simple method is to observe the number of iterations taken at each continuation step. By setting a desired target number of iterations, the method compares the actual number of iteration to the target. If the actual number is smaller, then the next step length can be a little larger than the previous one. On the flip side, if the actual number is greater, then the next step length should be a little smaller than the previous one. However this method fails to achieve the desired results if any control device is forced out of its normal operating region. For example, if an ULTC transformer model adjusts a tap setting to bring the controlled bus voltage within its specified tolerance, then it will take a few extra iterations for the Newton method to converge. These extra iterations would shorten the following predictor step, causing the predictor to take shorter steps, hence making it less efficient [10, 11].

The algorithm of conventional CPF is illustrated as flow chart in figure 2-4. The stop criterion of the flowchart as presented in figure 2-4 is a comparison of the current value of the load parameter λ with the previous one. If the current λ is bigger than the previous one, then the load parameter is still increasing and the computational steps need to continue. If the current λ is smaller than the previous one, then the maximum load parameter has been exceeded and the algorithm is terminated.

2.3 Summary

The details of implementation regarding to the parameterization, predictor-corrector scheme and step size control have been described. CPF can solve the power flow near or at the critical point without numerical ill-conditioning. The CPF has been developed and improved for the requirement of modern electric power systems, and many researchers and authors made great efforts in this area.

Chapter 3: Variations of Continuation Power Flow

This chapter introduces some variation work of the standard CPF. One is the enhanced CPF (ECPF) method, which employs a nonlinear predictor to control the step size; the other one is our proposed GCPF method, the details of GCPF algorithm procedures is described in this chapter.

3.1 Introduction

CPF is a classic method to calculate and plot PV curves. CPF incorporates the nonlinear relationship between loads and voltage magnitudes. The predictor-corrector scheme plays an important role in CPF methods [9, 10, 11, 29, 30]. In order to improve CPF methods, several techniques have been developed for the predictor-corrector scheme. The Homotopy method [63] and Ajjarapu-Christy method [11, 31, 72] make use of the tangent vector to compute the predicted solution, but the tangent vector predictor does not employ a large step size and can be time-consuming. The CPF method developed by Chiang [10] makes use of a secant-based vector as the predictor [73, 74], which is faster in predicting the solution than the tangent vector method.

3.2 Enhanced Continuation Power Flow

Mori and Yamada developed a nonlinear predictor method [30], this method improved the performance of CPF method by employing a nonlinear predictor instead of the tangent or secant predictor. The nonlinear predictor allows the algorithm to take a larger step-size than the linear one, the method makes use of the Lagrange Polynomial Interpolation Formula [55] to predict the solution, so that the error between predicted and actual solution is reduced. The step-size can be increased to speed up the computational time.

The nonlinear predictor is expected to give a predicted solution within a very close range of the actual solution, so that a larger step-size can be used. In this paper a rule based step-size control algorithm has also been proposed, which takes different step-sizes in different parts of the PV curve.

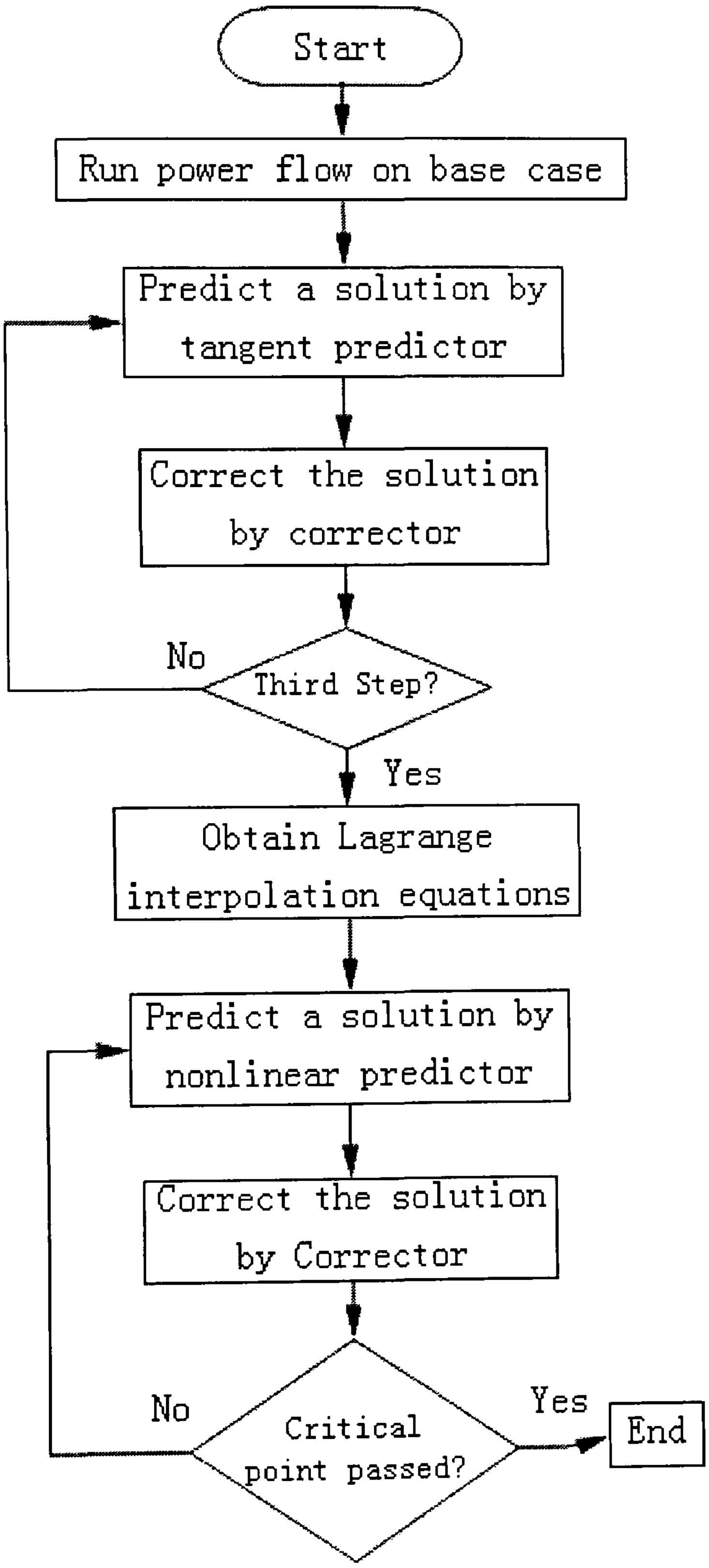

Start
Run power flow on base case
Predict a solution by tangent predictor
Correct the solution by corrector
No
Third Step?
Yes
Obtain Lagrange interpolation equations
Predict a solution by nonlinear predictor
Correct the solution by Corrector
No
Critical point passed?
Yes
End

Figure 3-1: Flowchart of CPF with nonlinear predictor

3.2.1 Nonlinear Predictor

A Lagrange polynomial interpolation formula [55] has been employed as the mathematical model for the nonlinear predictor. As we know, three known solutions can determine a quadratic equation [30]. Once the Lagrange Interpolation function is determined, we can predict the next solution from the fourth step. This algorithm is described by the flowchart in figure 3-1.

The stop criterion of the flowchart as presented in figure 3-1 is a comparison of the current value of the load parameter λ with the previous one. If the current λ is bigger than the previous one, then the load parameter is still increasing and calculation steps need to continue. If the current λ is smaller than the previous one, then the maximum load parameter has been exceeded and the algorithm is terminated.

3.2.2 Rule based step-size control

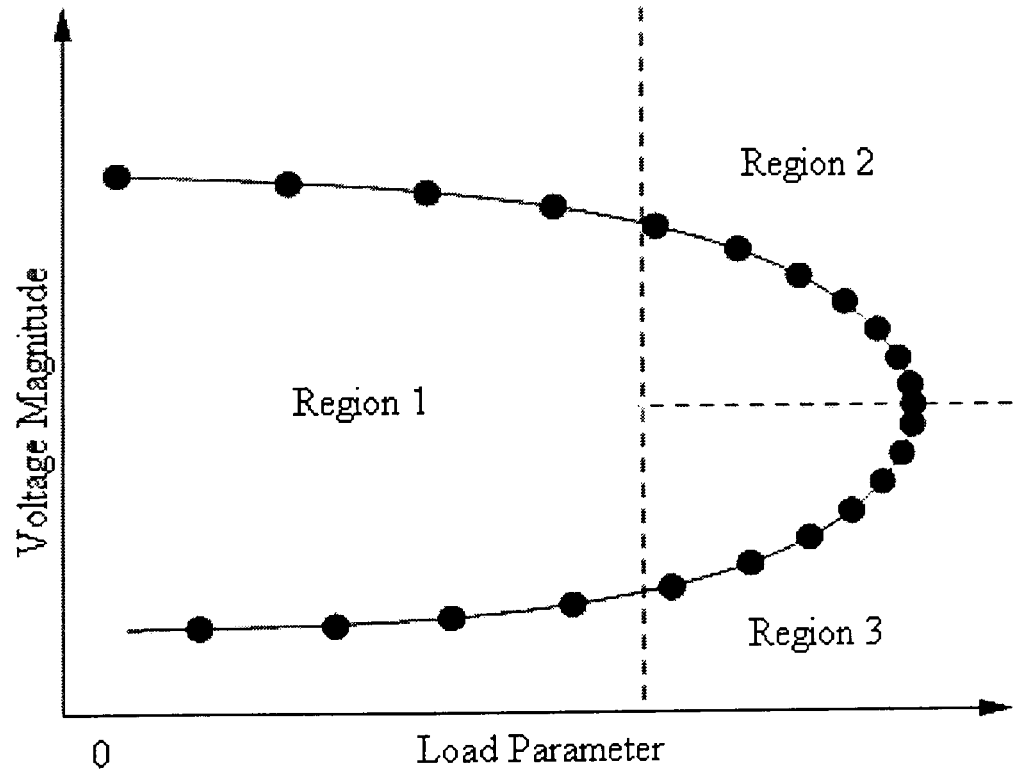


Figure 3-2: Decomposition of PV curve for Step-size Control

Mori and Yamada developed a rule-based algorithm for controlling step-size to speed up CPF computational time. In this proposed method, the PV curve is decomposed into three regions as illustrated in figure 3-2.

In region 1:

This part of curve is quite flat, so a large step-size can be taken. According to the predictor error [30], the step-size can be adjusted. If the error of the predictor exceeds a critical value, the predictor is forced to take a smaller step-size than the previous one.

In regions 2 and 3:

The step-size is changed according to the gradient of the PV curve, which means that when the continuation step is closer to the bifurcation point, the gradient value of $|\Delta x/\Delta \lambda|$ becomes larger, if $|\Delta x/\Delta \lambda|$ exceeds a certain pre-set value, the step-size should be reduced to a smaller value than the previous one. On the other hand, when the continuation step passes through the bifurcation point and the step enters the region 3, $|\Delta x/\Delta \lambda|$becomes smaller, then, the step-size would be enlarged to a value greater than the previous one.

This proposed method combines a nonlinear predictor with rule based step-size control, which improves the CPF efficiency. The nonlinear predictor can provide a more accurate solution predicting technique and rule based step-size control algorithm can skip some unnecessary continuation steps to speed up the computational time.

3.3 Q-limit Guided CPF Method

This paper proposes a novel Q-limit GCPF method [51] that accounts for reactive power limits and has been further developed in order to improve performance with regard to robustness and computational efficiency when compared with the CCPF method. The GCPF method is based on the standard CPF method, which offers a flexible and reliable solution procedure without suffering from divergence problems even when near the bifurcation point.

From equation (2-1), the reactive power generation Q_{G_i} can be defined as follows:

$$Q_{G_i} = v_i \sum_{j=1}^{n} y_{ij} v_j \sin(\delta_i - \delta_j - \alpha_{ij}) + Q_{L_i} \tag{3-1}$$

Where y_{ij} and α_{ij} are admittance matrix elements in polar form, Q_{L_i} denotes the reactive power demand at generator i and n is the total number of buses in the power system.

For a typical power system, the admittance matrix is known, so y_{ij} and α_{ij} are constants, and the voltage magnitude v_i of generator i is also a constant [67]. Assuming that a power system is operating under normal conditions, the phase angle differences $\delta_i - \delta_j$ are usually small. Therefore, the sinusoidal function can be approximated as a constant as follows:

$$\sin(\delta_i - \delta_j - \alpha_{ij}) \approx \sin(-\alpha_{ij}) \tag{3-2}$$

It can then be assumed that the voltage magnitude v_j is the only variable and that it has a strong relationship with Q_{G_i}. If we now consider the relationship of these two quantities and combine the constants as described above, we can simplify equation (3-1) as follows:

$$Q_{G_i} \approx Kv_j + Q_{L_i} \tag{3-3}$$

Where K is a linear coefficient matrix for the voltage magnitudes at the load buses. As we know, the load parameter λ has an almost quadratic relationship with voltage magnitude [10, 11, 51]. Consequently, the load parameter λ will also have an approximate quadratic relationship with respect to reactive power output of the generators:

$$\lambda = A_i v_i^2 + B_i v_i + C_i = a_i Q_{G_i}^2 + b_i Q_{G_i} + c_i \tag{3-4}$$

Lagrange interpolation is a good curve fitting technique that can be exploited to represent the quadratic relationship between λ and Q_{G_i}. Since quadratic behaviour can be defined by three points in a plane, accepting the initial state point, we require another two solution points along the curve to construct the full interpolation function.

If we have the data of three solution points equation (3-4) can then be defined. Then the extreme value of load parameter λ can be calculated by setting $d\lambda / dQ_{G_i}$ to zero when

$$q_i = -\frac{b_i}{2a_i} \tag{3-5-a}$$

$$\lambda_i = \frac{4a_i c_i - b_i^2}{4a_i} \tag{3-5-b}$$

(λ_i, q_i) is the extreme value point for the single Q_{G_i} - λ curve at bus i.

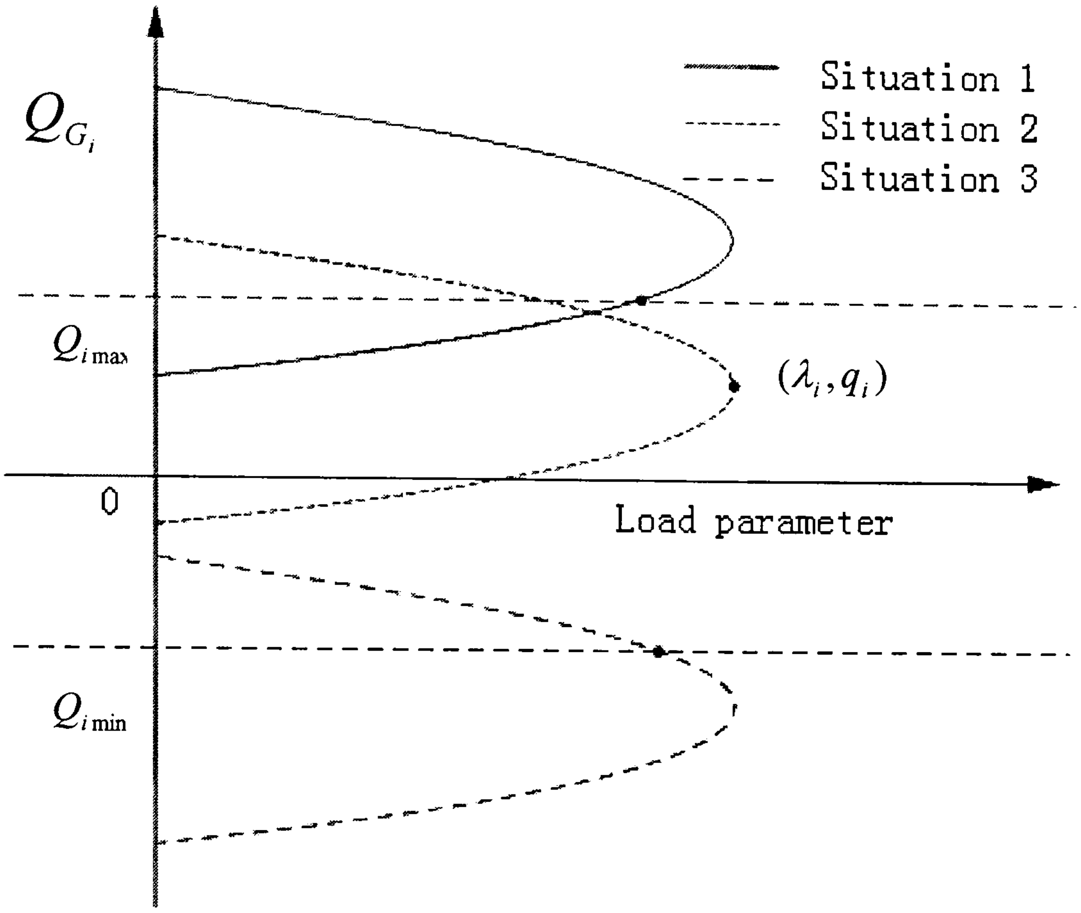


Figure 3-3: Extreme value points and Q-Limit Situations

We illustrated three situations, where Q_{G_i} may locate in figure 3-3. As the value of Q_{G_i} should lie between the initial value and the extreme value q_i. Situations 1 and 3 illustrate that Q_{G_i} will break its upper or lower Q-Limit, respectively. It is important to note that in situation 2, Q_{G_i} will not break its Q-limit. As we are only concerned with a possible limit breaking, we are only concerned with the generators that are prone to break their Q-limits

$$q_i \geq Q_{i\max} \tag{3-6-a}$$

$$\text{or } q_i \leq Q_{i\min} \tag{3-6-b}$$

because the other cases are within the system Q-limits.

Therefore, we can now substitute the $Q_{\min}$ and $Q_{\max}$ values of generators that are breaking Q-limits into the equation (3-4) as follows:

$$\lambda_{q\max}^{i} = a_i Q_{i\max}^2 + b_i Q_{i\max} + c_i \qquad (3\text{-}7\text{-}a)$$

$$\lambda_{q\min}^{i} = a_i Q_{i\min}^2 + b_i Q_{i\min}^2 + c_i \qquad (3\text{-}7\text{-}b)$$

Where $Q_{i\min}$ and $Q_{i\max}$ are the Q-limits of bus i. In case (3-6-a), we only consider upper limit and compute $\lambda_{q\max}^{i}$ from (3-7-a), alternatively in case (3-6-b) we only compute $\lambda_{q\min}^{i}$ from (3-7-b).

In each case, only the smallest value from equations (3-7-a) and (3-7-b) will be of interest. We can represent this as follows:

$$\lambda_q = \min(\lambda_{q\min}^{i}, \lambda_{q\max}^{i}) \qquad (3\text{-}8)$$

Subject to $\lambda_q > \lambda_0$, we take $\Delta s = \lambda_q - \lambda_0$ as the new step-size to guide the CPF method from the current point (x_0, λ_0) to the new operating point (x_q, λ_q). We then change Δs back to the initial value to carry on the CPF computation until the Q-limit breaking point is reached and then change the bus type from PV to PQ. We then set this Q-limit breaking point as the new starting point (x_0, λ_0) and the whole process is repeated again until the bifurcation point is reached.

It is important to note that it is possible for a Q-limit breaking point to be reached during the computation of the three consecutive solution points that are required for predicting λ_q; if this occurs the whole process is restarted at the new Q-limit breaking point and another three solution points are computed for the prediction process. If there are no generator buses satisfying equations (3-6-a) and (3-6-b), then no buses are breaking their Q-limits. Therefore, we take the smallest value of equation (3-5-b) as λ_q to guide the continuation steps until the bifurcation point is reached.

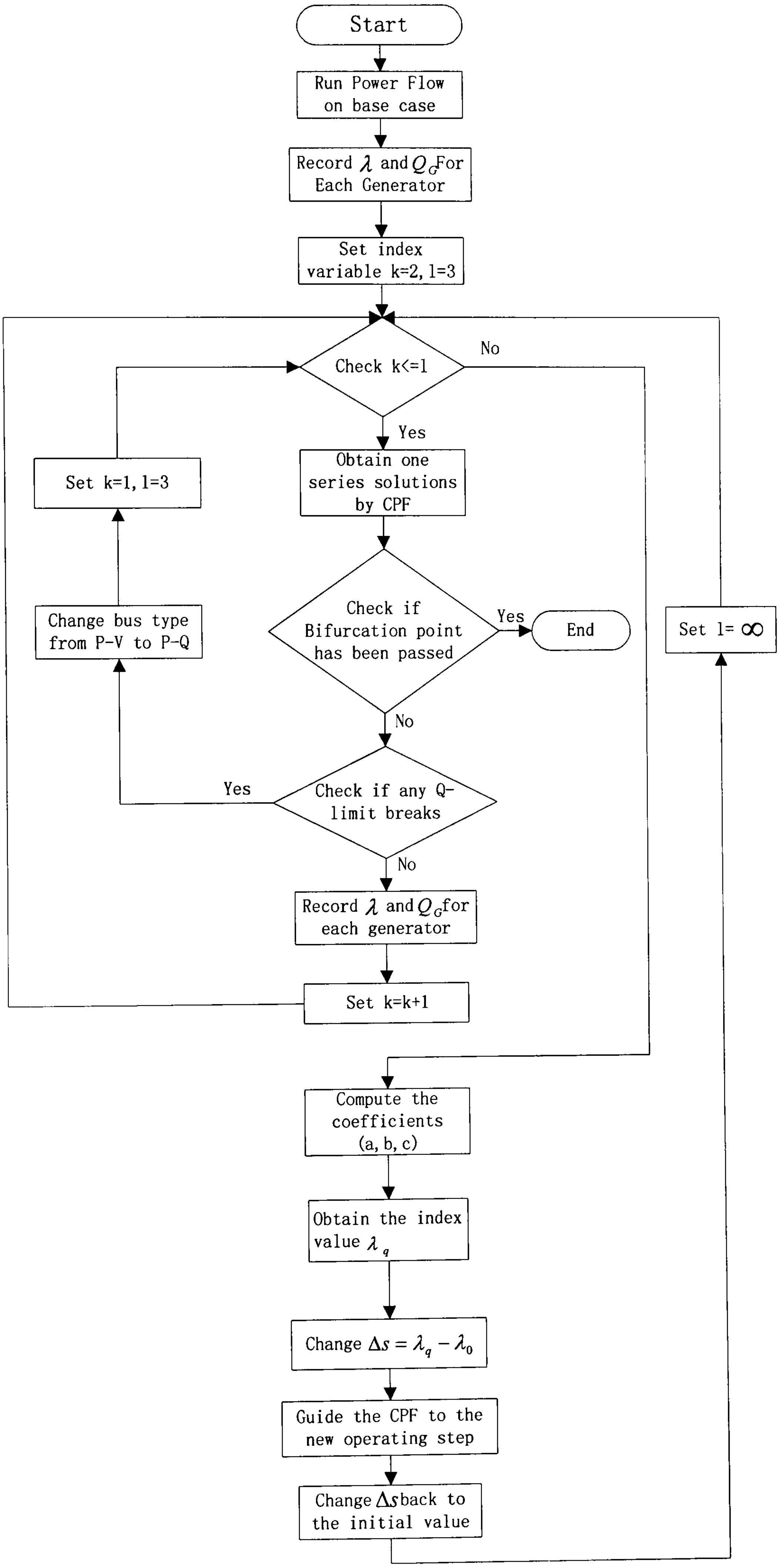

Start
Run Power Flow on base case
Record λ and Q_GFor Each Generator
Set index variable k=2, l=3
Check k<=l
No
Yes
Obtain one series solutions by CPF
Set k=1, l=3
Change bus type from P-V to P-Q
Check if Bifurcation point has been passed
Yes
End
Set l= ∞
No
Yes
Check if any Q-limit breaks
No
Record λ and $Q_G$for each generator
Set k=k+1
Compute the coefficients (a, b, c)
Obtain the index value λ_q
Change $\Delta s = \lambda_q - \lambda_0$
Guide the CPF to the new operating step
Change Δsback to the initial value

Figure 3-4: Flowchart of Novel Q-limit GCPF Algorithm

We can now represent the GCPF algorithm using the flow chart as presented in figure3-4. Where *k* and *l* are the index variables, such that *k* controls the data collection procedure across three consecutive solutions and *l* controls the number of continuation steps after the data collection procedure and up until the Q-limit breaking point is reached. Where *a*, *b* and *c* are the coefficients of equation (3-4), (3-7-a) and (3-7-b).

3.4 Summary

ECPF method makes use of a nonlinear predictor instead of a conventional linear tangent or secant predictor, the linear predictor employs a small step size in the CPF computing process, but the nonlinear predictor is not restricted to a small step size. In other words, the nonlinear predictor enables the CPF method to take a large step size than the linear case, which increases the computational speed of the CPF method. The feasible step size control mechanism is also employed in order to speed up the CPF computation. A large step size is used in the flat part of the solution curve and a small step size is used in the part with a high degree of curvature. This ECPF approach does improve the standard CPF method efficiently. But it is still necessary to improve the performance of CPF methods further in order to meet the requirements of present and future time-scales.

The proposed GCPF method is more efficient, which employs an original step size control rule that is fundamentally different from the conventional feasible step size as controlled by the gradient of the solution curve. The step size is controlled by calculating the distance of continuous Q-limit breaking points and we make use of a nonlinear Lagrange polynomial interpolation technique in order to predict at which point the reactive power limit of a generator is reached. The closest Q-limit breaking point is then taken as the next continuation step. In this manner the continuation steps are efficiently guided by the prediction of Q-limit points and as a consequence of the more adaptable step size control than the conventional rule based method, the GCPF method significantly reduces the number of algorithmic continuation steps, which results in saving the computational time.

Chapter 4: Applications of Linear Solvers in Power Flow

This chapter introduces the developmental history of the project of applications of iterative solvers in power flow calculation, and also presents the summary of AMD ordering scheme, ILU factorization method and linear solver approaches those have been investigated in my research work.

4.1 Introduction and Background

The power flow is a well known algorithmic problem which is usually solved by the Newton-Raphson method. The power flow function is a set of nonlinear equations, which could be solved by an appropriate linear solver after linearization. All the Newton-Raphson based power flow programmes have a common feature that one large or two small size linear equation systems have to be solved in each Newton-Raphson iterative step. We know that CPF is a process that computes many power flow solutions, which means the main work of CPF computation is solving linear equations continuously, which costs the most computational time, consequently, an efficient linear solver plays an important role in CPF calculation and it is another good way to speed up CPF method.

It is well known that the Gaussian elimination method is a very popular direct method in solving linear equations used in engineering system including power system. With the development of mathematical algorithms, some researchers have applied non-stationary iterative methods in power system to solve the large set of linear equations [34, 36, 37, 39]. Such as literature [34] described the first application of the Conjugate Gradients (CG) method [33, 75] to the decoupled power flow. In literature [36], another application of CG methods to a static security power flow problem is described. For fast decoupled power flow, these papers present a significant performance improvement of CG based methods compared to the direct method.

Because the CG method only can deal with positive definite and symmetric linear system matrices, which only could be applied to the decoupled power flow method. Generally speaking, the application of CG method only works when preconditioning is applied to the linear system of equations. Good preconditioning will group the eigenvalues of the transformed linear system matrix and will result in faster convergence. Fortunately, there is a natural fit between the preconditioned CG method and the decoupled power flow method, because the decoupled power flow has constant linear system matrices. Thus, for every power flow calculation this preconditioning matrix must be computed only once and remains constant for all Newton-Raphson steps. Actually, only good pre-conditioners allow an efficient implementation of CG methods to linear equation systems of power flow.

In literature [36], a derivation of the "Krylov subspace power flow methodology" applied to the power flow problem is given to introduce power system application developers to the mathematical problem. The main distinction to the CG methods is in the fact that the "Krylov subspace power flow methodology" is also applicable to asymmetric, indefinite linear system matrices. Krylov subspace power flow (KSPF) does not need any explicit computation of the Jacabian terms during the iterations and power flow steps, this method can be employed directly to solve the linear equations of the Newton-Raphson power flow method, due to the characters of Jacobian matrix that is an asymmetric, possibly indefinite matrix [47].

Mathematicians have developed several methods to solve this type of linear system of equations, for example the generalized minimum residual method (GMRES), bi-conjugate gradient method (BiCG) and bi-conjugate gradient stabilized method (BiCGStab) are distinctly different methods for the solution of this class of problems. The full details of these methods have been described in the following sections.

The GMRES method is the generalization of the CG algorithm for asymmetric and indefinite linear system matrices. These two methods have in common that the solution error decreases from one iteration to the next and the actual solution is obtained within a given maximum number of iterations. The other iterative methods use combinations of CG concepts and heuristics to obtain a solution for the linear system. As a consequence the residual of these methods is not guaranteed to decrease during the iterations.

Solving a linear equation of the form Ax = b plays an important role in power flow calculation, and takes the most computational time in a power flow computation. It is well established that iterative methods for solving large sparse sets of linear equations can naturally profit from parallelization and factorization [39, 45, 46, 91]. So, I think that is a good project to explore an efficient linear solver to speed up CPF computation. In this thesis, I have made some meaningful comparisons between the performance of the direct and iterative methods.

The direct method investigated in this research is the Gaussian Elimination method [76]. A direct method solves the linear equations in a finite number of steps and may require extensive computational storage and arithmetic processing. Direct methods have been improved over the years through the use of efficient ordering techniques and sparse programming [40-44]. Although, the computation time of a direct method can still be the limiting factor in cases involving large-scale equation systems or numerous repeated solutions [34] such as CPF computation. Whatever, the direct method is the most commonly used as the linear solver in power flow computation.

It has been reported that the application of some iterative methods in power flow calculation decreases the computation time, compared with the LDU factorization based direct methods, for large power system problems [34, 35]. In this research, the incomplete LU factorization (ILU) technique is used as a pre-conditioner to accelerate the GMRES, BiCG and BiCGStab methods, which are quite effective in clustering the eigenvalues of the linear system [47].

4.2 Direct Method

The direct method investigated in this research is the Gaussian Elimination method [76], which is a well know and common used algorithm for linear system of equations on the form of $Ax = b$. The process of Gaussian elimination has two parts. The first part (Forward Elimination) reduces a given system to either triangular or echelon form, or results in a degenerate equation with no solution, indicating the system has no solution. This is accomplished through the use of elementary row operations. The second part uses back substitution to find the solution of the system above.

Stated equivalently for matrices, the first part reduces a matrix to row echelon form using elementary row operations while the second reduces it to reduced row echelon form, or row canonical form.

Another point of view, which turns out to be very useful to analyze the algorithm is that Gaussian elimination computes a matrix decomposition. The three elementary row operations used in the Gaussian elimination (multiplying rows, switching rows, and adding multiples of rows to other rows) amount to multiplying the original matrix with invertible matrices from the left. The first part of the algorithm computes an LU decomposition, while the second part writes the original matrix as the product of a uniquely determined invertible matrix and a uniquely determined reduced row-echelon matrix.

Because the widely application of this method, I didn't describe the details of the Gaussian Elimination method in this section.The detailed algorithm is presented in the literature [76].

4.3 Iterative Methods

The term "iterative method" refers to a wide range of techniques that use successive approximations to obtain more accurate solutions to a linear system at each step. At present, there are two types of iterative methods. Stationary methods are older, simpler to understand and implement, but usually not as effective. Nonstationary methods are a relatively recent development; their analysis is usually harder to understand, but they can be highly effective.

The rate at which an iterative method converges depends greatly on the spectrum of the coefficient matrix. Hence, iterative methods usually involve a second matrix that transforms the coefficient matrix into one with a more favorable spectrum. The transformation matrix is called a pre-conditioner. A good pre-conditioner improves the convergence of the iterative method [37, 49], sufficiently to overcome the extra cost of constructing and applying the pre-conditioner. Indeed, without a pre-conditioner the iterative method may even fail to converge.

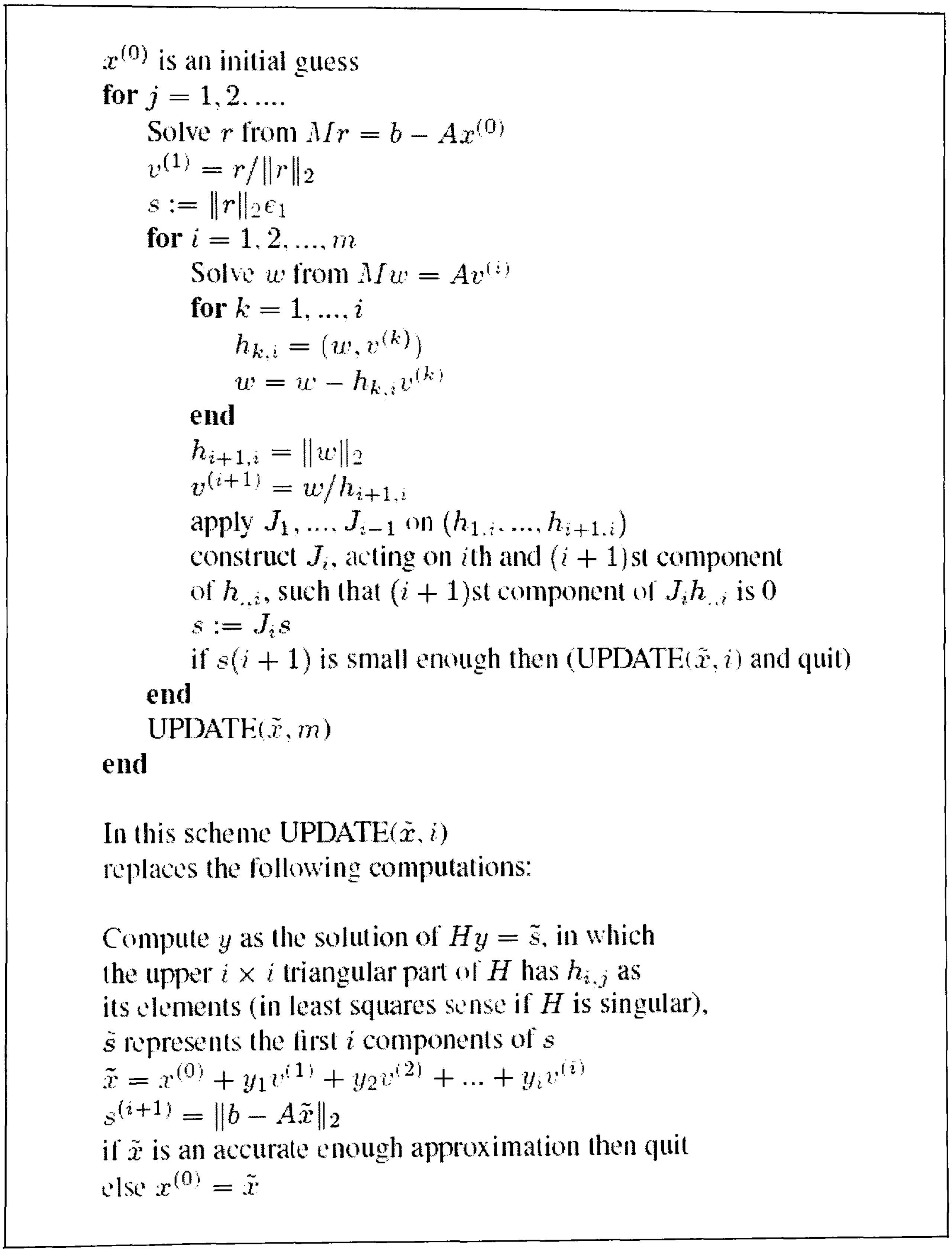
4.3.1 Generalized Minimum Residual Method (GMRES)

The Generalized Minimal Residual method [48, 77] is an extension of MINRES (Minimal Residual method) [78] (which is only applicable to symmetric systems) to asymmetric systems. It generates a sequence of orthogonal vectors, but in the absence of symmetry this can no longer be done with short recurrences; instead, all previously computed vectors in the orthogonal sequence have to be retained. For this reason, "restarted" versions of the method are used.

In the Conjugate Gradient method, the residuals form an orthogonal basis for the space $span\{r^{(0)}, Ar^{(0)}, A^2r^{(0)}, \cdots\}$. In GMRES, this basis is formed explicitly:

$$\omega^{(i)} = A v^{(i)}$$
$$\text{for } k = 1, \cdots, i$$

```
        $\omega^{(i)} = \omega^{(i)} - (\omega^{(i)}, v^{(k)})v^{(k)}$
    end
        $v^{(i+1)} = \omega^{(i)} / \|\omega^{(i)}\|$
```



```
$x^{(0)}$ is an initial guess
for $j = 1, 2, ....$
    Solve $r$ from $Mr = b - Ax^{(0)}$
    $v^{(1)} = r/\|r\|_2$
    $s := \|r\|_2 e_1$
    for $i = 1, 2, ..., m$
        Solve $w$ from $Mw = Av^{(i)}$
        for $k = 1, ..., i$
            $h_{k,i} = (w, v^{(k)})$
            $w = w - h_{k,i} v^{(k)}$
        end
        $h_{i+1,i} = \|w\|_2$
        $v^{(i+1)} = w/h_{i+1,i}$
        apply $J_1, ..., J_{i-1}$ on $(h_{1,i}, ..., h_{i+1,i})$
        construct $J_i$, acting on $i$th and $(i+1)$st component
        of $h_{.,i}$, such that $(i+1)$st component of $J_i h_{.,i}$ is 0
        $s := J_i s$
        if $s(i+1)$ is small enough then (UPDATE($\tilde{x}, i$) and quit)
    end
    UPDATE($\tilde{x}, m$)
end
```

In this scheme UPDATE($\tilde{x}, i$)
replaces the following computations:

```
Compute $y$ as the solution of $Hy = \tilde{s}$, in which
the upper $i \times i$ triangular part of $H$ has $h_{i,j}$ as
its elements (in least squares sense if $H$ is singular),
$\tilde{s}$ represents the first $i$ components of $s$
$\tilde{x} = x^{(0)} + y_1 v^{(1)} + y_2 v^{(2)} + ... + y_i v^{(i)}$
$s^{(i+1)} = \|b - A\tilde{x}\|_2$
if $\tilde{x}$ is an accurate enough approximation then quit
else $x^{(0)} = \tilde{x}$
```

Figure 4-1: The Preconditioned GMRES(m) Method

The reader may recognize this as a modified Gram-Schmidt orthogonalization. Applied to the Krylov sequence $\{A^k r^{(0)}\}$ this orthogonalization is called the "Arnoldi method" [79]. The inner product coefficients $(\omega^{(i)}, v^{(k)})$ and $\|\omega^{(i)}\|$ are stored in an upper Hessenberg matrix.

The GMRES iterates are constructed as

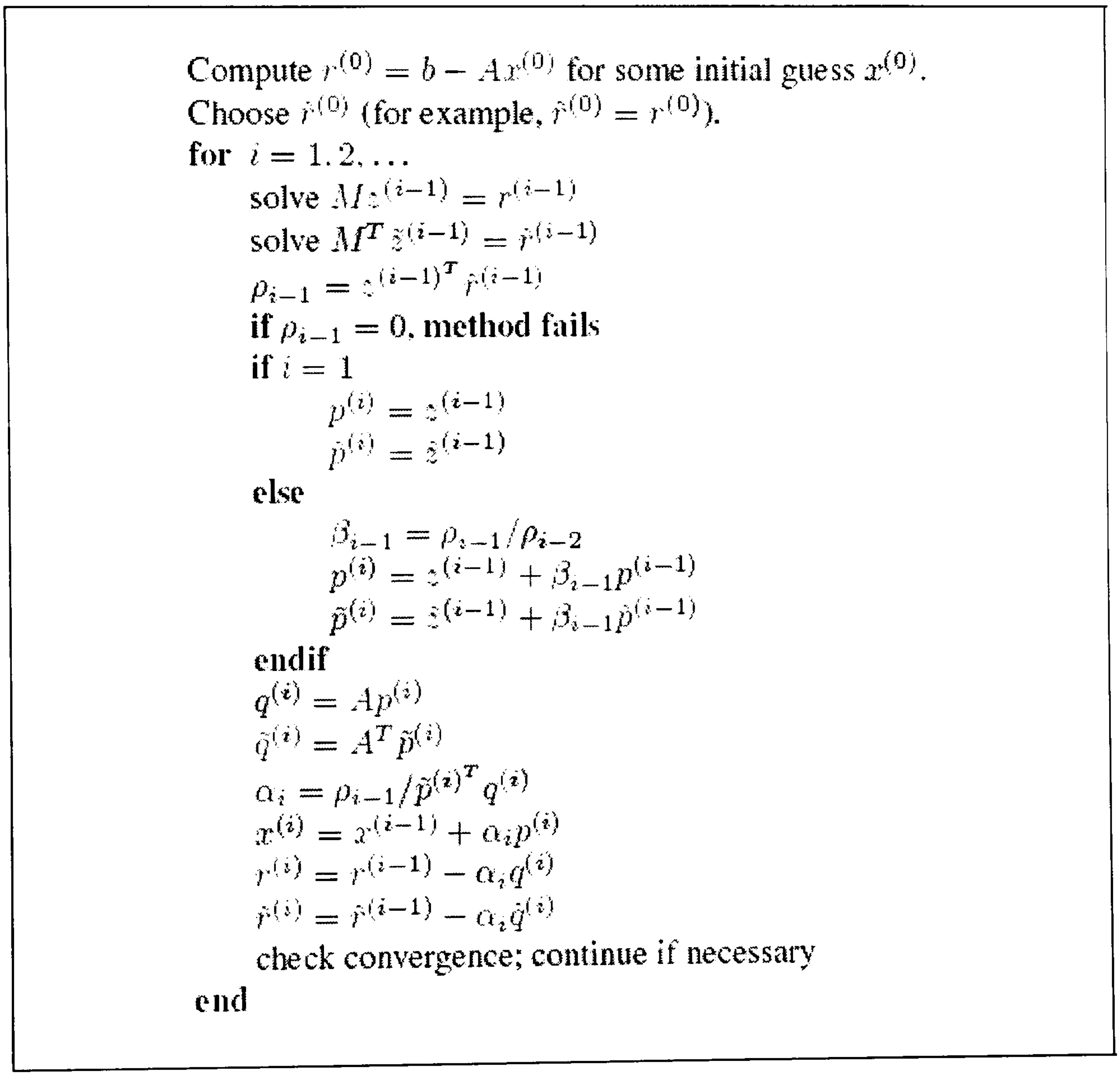
$$x^{(i)} = x^{(0)} + y_1 v^{(1)} + \cdots + y_i v^{(i)}$$

where the coefficients y_k have been chosen to minimize the residual norm $\|b - Ax^{(i)}\|$. The GMRES algorithm has the property that this residual norm can be computed without the iterate having been formed. Thus, the expensive action of forming the iterate can be postponed until the residual norm is deemed small enough.

The technical description for the restarted GMRES(m) algorithm with pre-conditioner M is presented in figure 4-1. The full detail of GMRES hasn't been described in this section because it is a quite well known method. The detailed GMRES method has been presented in the literature [48].

4.3.2 Bi-conjugate Gradient Method (BiCG)

The Conjugate Gradient method is not suitable for asymmetric systems because the residual vectors cannot be made orthogonal with short recurrences. The GMRES method retains orthogonality of the residuals by using long recurrences, at the cost of a larger storage demand. The Bi-conjugate Gradient method [49] takes another approach, replacing the orthogonal sequence of residuals by two mutually orthogonal sequences, at the price of no longer providing a minimization.



```
Compute $r^{(0)} = b - Ax^{(0)}$ for some initial guess $x^{(0)}$.
Choose $\tilde{r}^{(0)}$ (for example, $\tilde{r}^{(0)} = r^{(0)}$).
for $i = 1, 2, \ldots$
    solve $Mz^{(i-1)} = r^{(i-1)}$
    solve $M^T\tilde{z}^{(i-1)} = \tilde{r}^{(i-1)}$
    $\rho_{i-1} = z^{(i-1)^T}\tilde{r}^{(i-1)}$
    if $\rho_{i-1} = 0$, method fails
    if $i = 1$
        $p^{(i)} = z^{(i-1)}$
        $\tilde{p}^{(i)} = \tilde{z}^{(i-1)}$
    else
        $\beta_{i-1} = \rho_{i-1}/\rho_{i-2}$
        $p^{(i)} = z^{(i-1)} + \beta_{i-1}p^{(i-1)}$
        $\tilde{p}^{(i)} = \tilde{z}^{(i-1)} + \beta_{i-1}\tilde{p}^{(i-1)}$
    endif
    $q^{(i)} = Ap^{(i)}$
    $\tilde{q}^{(i)} = A^T\tilde{p}^{(i)}$
    $\alpha_i = \rho_{i-1}/\tilde{p}^{(i)^T}q^{(i)}$
    $x^{(i)} = x^{(i-1)} + \alpha_i p^{(i)}$
    $r^{(i)} = r^{(i-1)} - \alpha_i q^{(i)}$
    $\tilde{r}^{(i)} = \tilde{r}^{(i-1)} - \alpha_i \tilde{q}^{(i)}$
    check convergence; continue if necessary
end
```

Figure 4-2: The Preconditioned Bi-conjugate Gradient Method

The update relations for residuals in the Conjugate Gradient method are augmented in the BiConjugate Gradient method by relations that are similar but based on A^T instead of A. Thus we update two sequences of residuals

$$r^{(i)} = r^{(i-1)} - \alpha_i A p^{(i)}, \quad \tilde{r}^{(i)} = \tilde{r}^{(i-1)} - \alpha_i A^T \tilde{p}^{(i)}$$

and two sequences of search directions

$$p^{(i)} = r^{(i-1)} + \beta_{(i-1)} p^{(i-1)}, \quad \tilde{p}^{(i)} = \tilde{r}^{(i-1)} + \beta_{(i-1)} \tilde{p}^{(i-1)}$$

The choices

$$\alpha_i = \frac{\tilde{r}^{(i-1)^T} r^{(i-1)}}{\tilde{p}^{(i)^T} A p^{(i)}}, \quad \beta_i = \frac{\tilde{r}^{(i)^T} r^{(i)}}{\tilde{r}^{(i-1)^T} r^{(i-1)}}$$

ensure the bi-orthogonality relations

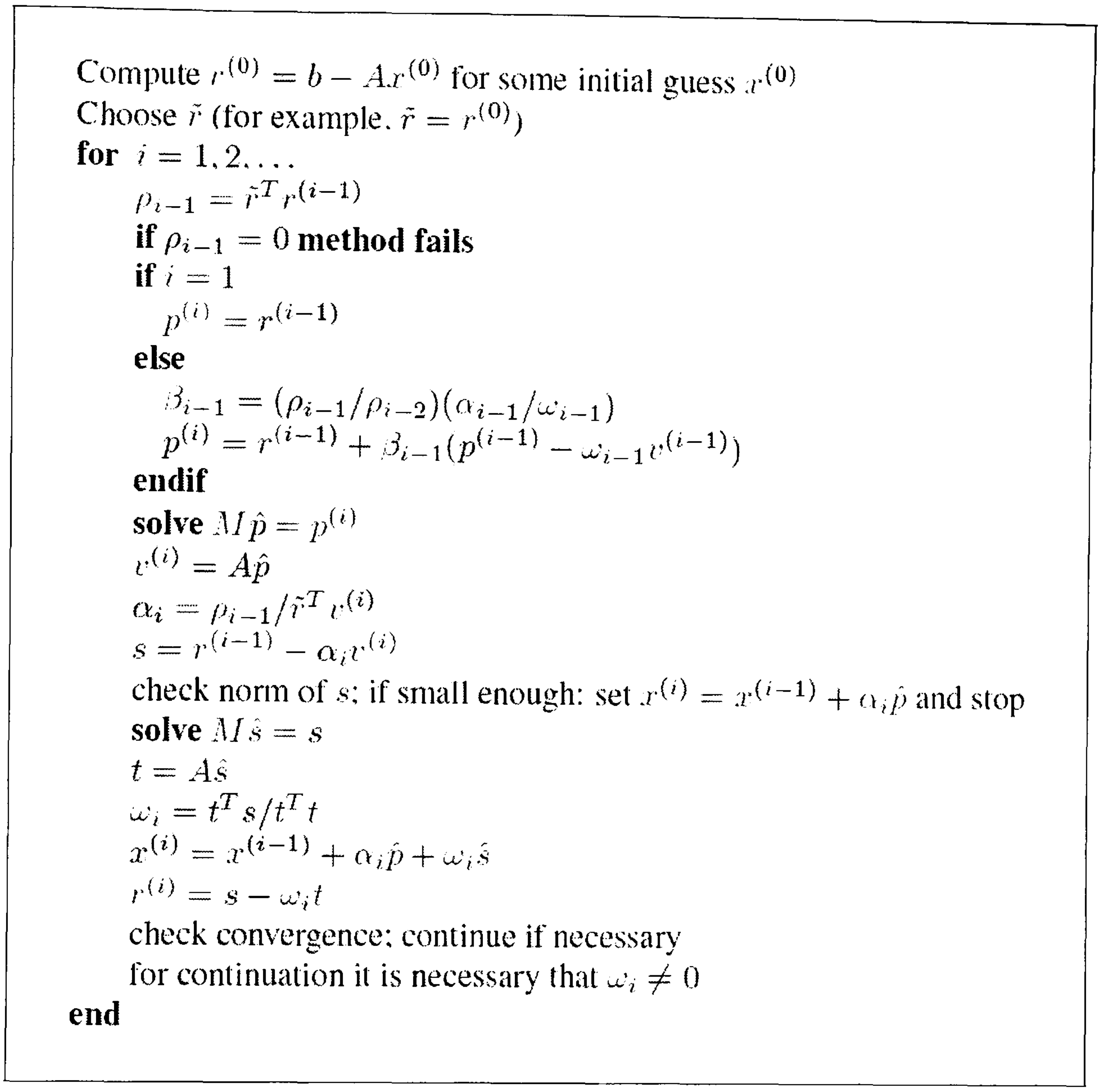
$$\tilde{r}^{(i)^T} r^{(j)} = \tilde{p}^{(i)^T} A p^{(j)} = 0 \qquad \text{if } i \neq j$$

The algorithm description for the preconditioned BiCG method with pre-conditioner M is presented in Figure 4-2. The full detail of BiCG hasn't been described in this section because it is a quite well known method. The detailed BiCG method has been presented in the literature [49].

4.3.3 Bi-Conjugate Gradient Stabilized Method (BiCGStab)

The Bi-Conjugate Gradient Stabilized method (BiCGStab) [50] was developed to solve unsymmetric linear systems while avoiding the often irregular convergence patterns of the Conjugate Gradient Squared method [80]. Instead of computing the CGS sequence $i \mapsto P_i^2(A)r^{(0)}$, Bi-CGSTAB computes $i \mapsto Q_i(A)P_i(A)r^{(0)}$ where Q_i is an i th degree polynomial describing a steepest descent update.

The algorithm description for the preconditioned BiCGStab method with pre-conditioner M is given in Figure 4-3. The full detail of BiCGStab method hasn't been described in this section because it is a quite well known method. The detailed BiCGStab method has been presented in the literature [50].



```
Compute $r^{(0)} = b - Ax^{(0)}$ for some initial guess $x^{(0)}$
Choose $\tilde{r}$ (for example, $\tilde{r} = r^{(0)}$)
for $i = 1, 2, \ldots$
    $\rho_{i-1} = \tilde{r}^T r^{(i-1)}$
    if $\rho_{i-1} = 0$ method fails
    if $i = 1$
        $p^{(i)} = r^{(i-1)}$
    else
        $\beta_{i-1} = (\rho_{i-1}/\rho_{i-2})(\alpha_{i-1}/\omega_{i-1})$
        $p^{(i)} = r^{(i-1)} + \beta_{i-1}(p^{(i-1)} - \omega_{i-1}v^{(i-1)})$
    endif
    solve $M\hat{p} = p^{(i)}$
    $v^{(i)} = A\hat{p}$
    $\alpha_i = \rho_{i-1}/\tilde{r}^T v^{(i)}$
    $s = r^{(i-1)} - \alpha_i v^{(i)}$
    check norm of $s$; if small enough: set $x^{(i)} = x^{(i-1)} + \alpha_i \hat{p}$ and stop
    solve $M\hat{s} = s$
    $t = A\hat{s}$
    $\omega_i = t^T s/t^T t$
    $x^{(i)} = x^{(i-1)} + \alpha_i \hat{p} + \omega_i \hat{s}$
    $r^{(i)} = s - \omega_i t$
    check convergence; continue if necessary
    for continuation it is necessary that $\omega_i \neq 0$
end
```

Figure 4-3: The Preconditioned Bi-Conjugate Gradient Stabilized Method

4.4 Pre-conditioner for Iterative Methods

The convergence rate of iterative methods depends on spectral properties of the coefficient matrix. Hence one may attempt to transform the linear system into one that is equivalent in the sense that it has the same solution, but that has more favorable spectral properties. A pre-conditioner is a matrix that effects such a transformation [81, 82].

For instance, if a matrix M approximates the coefficient matrix A in some way, the transformed system

$$M^{-1}Ax = M^{-1}b$$

has the same solution as the original system $Ax = b$, but the spectral properties of its coefficient matrix $M^{-1}A$ may be more favorable.

Since using a pre-conditioner in an iterative method incurs some extra cost, both initially for the setup, and per iteration for applying it, there is a trade-off between the cost of constructing and applying the pre-conditioner, and the gain in convergence speed [83-86]. Certain pre-conditioners need little or no construction phase at all, but for others, such as incomplete factorizations, there can be substantial work involved. Although the work in scalar terms may be comparable to a single iteration, the construction of the pre-conditioner may not be vectorizable or parallelizable even if application of the pre-conditioner is. In that case, the initial cost has to be amortized over the iterations or over repeated use of the same pre-conditioner in multiple linear systems.

Iterative method convergence rate is strongly dependent on the coefficient matrix. In fact, the matrix may have a very large range of eigenvalues, causing poor convergence behaviour or even leading to divergence. When using pre-conditioners, a new matrix system is produced with eigenvalues closer to unity. For example, if B = LU is the pre-conditioner matrix, then the $L^{-1}AU^{-1}$matrix is closer to the identity matrix than A.

Therefore, the equation system $\left[L^{-1}AU^{-1}\right]*\left[Ux\right]=L^{-1}b$ is expected to convergence faster. Usually, the matrix B is derived by using incomplete LU decomposition of A.

4.4.1 Column Approximate Minimum Degree Ordering Scheme

Sparsity preserving ordering scheme is a crucial part of pre-conditioner, ordering affects the eigenvalue spectrum efficiently, which in turn has an impact on convergence rates of the pre-conditioned iterative methods. The iteration number of iterative methods is almost directly related to the norm of the remainder matrix R ($A=M+R$,where M is the pre-conditioner, A is the coefficient matrix and R is the remainder) [87].

In the direct method, the ordering scheme can minimize the storage and the number of floating-point operations. Although sparsity is preserved, ordering may affect the size of the norm of R. A good ordering scheme is aimed at making $M^{-1}A$ to be close to the identify matrix [87]: Ideally the spectrum of $M^{-1}A$ will have a single eigenvalue, in practice we content ourselves with a spectrum containing a small number of clusters of eigenvalues.

Linear solver computes the factorization **PAQ** = **LU** of a sparse matrix **A**, where **P** and **Q** are permutation matrices, **L** is a lower triangular matrix, and **U** is an upper triangular matrix. The row ordering **P** is selected during factorization using standard partial pivoting with row interchanges. The goal is to select a column preordering, **Q**, based solely on the nonzero pattern of **A**, that limits the worst-case number of nonzeros in the factorization.

The fill-in also depends on **P**, but **Q** is selected to reduce an upper bound on the fill-in for any subsequent choice of **P**. The choice of **Q** can have a dramatic impact on the number of nonzeros in **L** and **U**. One scheme for determining a good column ordering for **A** is to compute a symmetric ordering that reduces fill-in in the Cholesky factorization of $A^T A$. A conventional minimum degree ordering algorithm would require the sparsity structure of $A^T A$ to be computed, which can be expensive both in terms of space and time since $A^T A$ may be much denser than **A**. An alternative is to compute **Q** directly from the sparsity structure of **A**. The column approximate minimum degree (AMD) ordering scheme is based on the same strategy but uses a better ordering heuristic. AMD ordering scheme is faster and computes better orderings, with fewer nonzeros in the factors of the matrix. The full details of AMD ordering scheme algorithm is described in the literature [56].

4.4.2 Incomplete factorization

The pre-conditioner applied in this research is based on incomplete factorizations of the coefficient matrix. We call a factorization incomplete if during the factorization process certain *fill* elements, nonzero elements in the factorization in positions where the original matrix had a zero, have been ignored. Such a pre-conditioner is then given in factored form $M = LU$ with L being lower and U upper triangular. The efficacy of the pre-conditioner depends on how well M^{-1} approximates A^{-1}.

An important consideration for incomplete factorization pre-conditioners is the cost of the factorization process. Even if the incomplete factorization exists, the number of operations involved in creating it is at least as much as for solving a system with such a coefficient matrix, so the cost may equal that of one or more iterations of the iterative method. On parallel computers this problem is aggravated by the generally poor parallel efficiency of the factorization.

Such factorization costs can be amortized if the iterative method takes many iterations, or if the same pre-conditioner will be used for several linear systems, for instance in successive time steps or Newton iterations.

Incomplete factorizations can be given in various forms. If $M = LU$ (with L and U nonsingular triangular matrices), solving a system $Mx = y$ proceeds in the usual way below.

```
Let $M = LU$ and $y$ be given.
for $i = 1, 2, \ldots$
    $z_i = \ell_{ii}^{-1}(y_i - \sum_{j<i} \ell_{ij} z_j)$
for $i = n, n-1, n-2, \ldots$
    $x_i = u_{ii}^{-1}(z_i - \sum_{j>i} u_{ij} x_j)$
```

but often incomplete factorization are given as $M = (D+L)D^{-1}(D+U)$ (with D diagonal, and L and U now strictly triangular matrices, determined through the factorization process). In that case, one could use either of the following equivalent formulations for:

$$(D+L)z = y,\ (I + D^{-1}U)x = z \text{ or } (I + LD^{-1})z = y,\ (D+U)x = z$$

In either case, the diagonal elements are used twice (not three times as the formula for M would lead one to expect), and since only divisions with D are performed, storing D^{-1} explicitly is the practical thing to do. At the cost of some extra storage, one could store LD^{-1} or $D^{-1}U$, thereby saving some computation. Solving a system $Mx = y$ with $M = (D+L)D^{-1}(D+U)$ = $(D+L)(I+D^{-1}U)$ using the first formulation is described below.

```
Let $M = (D + L)(I + D^{-1}U)$ and $y$ be given.
for $i = 1, 2, \ldots$
    $z_i = d_{ii}^{-1}(y_i - \sum_{j<i} \ell_{ij} z_j)$
for $i = n, n-1, n-2, \ldots$
    $x_i = z_i - d_{ii}^{-1} \sum_{j>i} u_{ij} x_j$
```

4.5 Summary

The use of good pre-conditioner can improve the convergence of the iterative method, although there is a cost for constructing and applying the pre-conditioner. Since applying a pre-conditioner has extra cost, there is a balance between the cost of constructing and applying the pre-conditioner and the profit in convergence speed or the value of turning a divergence system into a convergent one.

How the AMD ordering scheme affects eigenvalue spectrum and convergence rate in dealing with the large sparse coefficient matrix has been discussed in the chapter seven.

In order to make meaningful comparisons between direct and iterative methods it is necessary to make sure we are having the same general testing conditions for all the methods. One critical issue is pre-conditioner, so we employed incomplete LU factorization with the AMD ordering scheme to all of the linear solvers including direct Gaussian Elimination method. The fully numerical comparison results and analysis have been presented in Chapter seven.

Chapter 5: Simulation Tool

In this research work, the simulation tool is based on the code of Matpower that is created by Power Systems Engineering Research Centre (PSERC) at Cornell University. I have made plenty of work on the modification and development of the original code of Matpower to simulate many experiments in algorithmic comparison between CCPF, ECPF and GCPF. I have also modified the code of Matpower to implement Newton-Raphson power flow using different linear solvers.

5.1 Introduction of Matpower

Matpower [57] is a package of m-files for solving power flow and optimal power flow problems. It is intended as a simulation tool for researchers and educators that are easy to use and modify. The code is designed to give the best performance possible while keeping the code simple to understand and modify.

Matpower has three power flow solvers. The default power flow solver is based on a standard Newton-Raphson method [88] using a full Jacobian, updated at each iteration. This method is described in detail in many textbooks and is popular used all over the world. The other two power flow solvers are variations of the fast-decoupled method [89]. Matpower implements the XB and BX variations as described in [90].

In Matpower, if any generator reactive power limit is violated after the computation of power flow, the corresponding bus is converted to a PQ bus, with the reactive output set to the limit, and the power flow computation repeats. The voltage magnitude at the bus will deviate from the specified value in order to satisfy the reactive power limit. If the generator at the reference bus reaches a reactive power limit and the bus is converted to a PQ bus, the first remaining PV bus will be used as the slack bus for the next iteration. This may result in the real power output at this generator being slightly off from the specified values.

Matpower uses an options vector to control the many options available, such as power flow algorithm and power flow termination criterion. It is similar to the options vector produced by the Matlab's Optimization Toolbox. The primary difference is that modifications can be made by the option name.

5.2 Modification and Development of Matpower

The Matpower provides the best performance to solve the power flow problem using Newton's method. Matpower takes advantage of Matlab's built-in sparse matrix, by which the computation of power flow becomes faster and less iterations. Therefore, the Matpower is a really useful and important tool for the computation of power systems.

What we are interested is to realize the continuation power flow based on the Matpower. Matpower's power flow solution just can calculate a single steady-state of a power system, but the continuation power flow is the process that needs to compute continuous solutions of power flow with the load demand and generation variations. So there are some modifications that need to be done according to the continuation power flow.

Q-limit guided continuation power flow method has five basic elements:

1. Parameterization
2. Predictor
3. Corrector
4. Step-size choose
5. Q-limit index

The procedure of modification and development of matpower follow the five basic elements.

5.2.1 Parameterize the Power Flow Equations

According to the CPF we described the details in Chapter two, there are three different types of parameterization: physical parameterization, local parameterisation and arc-length parameterization. In this research work we choose the third one, the arc-length parameterization to guide the modification of Matpower.

In this case, the reformulated power flow equations are the combination of the step length Δs and the original power flow equations:

$$\begin{cases} f(x) - \lambda b = 0 \\ \sum_{i=1}^{n} \left\{ (x_i - x_i(s))^2 + (\lambda - \lambda(s))^2 \right\} = \Delta s^2 \end{cases} \tag{5-1}$$

this is the continuation power flow mathematical model.

Lots of modification and creation in the part of Newton power flow function of Matpower needed with regard to the new CPF model. CPF brings one more unknown variable, the load parameter λ, in power flow, which means that we have to solve the load parameter λ at each solution. To realize this procedure we need to insert one column and one row elements into some matrix to enlarge the dimension, for example the Jacobian matrix if the dimension of Jacobian coefficient matrix is $n \times n$, it will be changed to $(n+1) \times (n+1)$.

The detail about the modification of Matpower to solve the combination equations 5-1 is described in the corrector section.

5.2.2 M-file of Predictor

We develop a new function m file in Matpower to serve as the predictor. The tangent method is chosen in GCPF algorithm as a predictor, the set of predictor equations is:

$$\begin{cases} f_{x_{n+1}} \dfrac{dx_i}{ds} + f_{x_{n+1}} \dfrac{d\lambda}{ds} = 0 \\ (\dfrac{dx_1}{ds})^2 + \cdots + (\dfrac{dx_n}{ds})^2 + (\dfrac{d\lambda}{ds})^2 = 1 \end{cases} \tag{5-2}$$

this is a set of n+1 equations in n+1 variables, the load parameter λ is the $(n+1)^{th}$ variable, please note that the first equation is a set of linear equations in its n+1 unknowns and the second equation is nonlinear.

A special method is presented below to solve this n+1 equation:

Suppose

$$dx_k / ds \neq 0, \quad 1 \leq k \leq n+1$$

let Df_k be the matrix of Df with the last column taken out, which is related with load parameter λ.

$$Df_k = \begin{bmatrix} \dfrac{\partial f_1}{\partial x_1} & \cdots & \dfrac{\partial f_n}{\partial x_n} \\ \vdots & \vdots & \vdots \\ \dfrac{\partial f_n}{\partial x_1} & \cdots & \dfrac{\partial f_n}{\partial x_n} \end{bmatrix} \tag{5-3}$$

suppose Df_k is not singular, so the unknowns below can be solved.

$$\frac{dx_1}{ds},\cdots\frac{dx_n}{ds} \tag{5-4}$$

in another form of

$$\frac{dx_i}{ds}=\beta_i\frac{d\lambda}{ds},\quad i=1,\cdots,n \tag{5-5}$$

the solution for the coefficients β_i can be obtained by applying Gaussian elimination to the matrix Df_k. Once the β_i is calculated, substitute (5-5) into the second equation of (5-2), the $\frac{d\lambda}{ds}$ can be solved by:

$$(\frac{d\lambda}{ds})^2=(1+\sum_{i=1}^{n}\beta_i^{\,2})^{-1} \tag{5-6}$$

the combination of (5-5) and (5-6) can solve all the tangent vector of continuation point (x^j,λ^j), a predictor step can be accomplished by integrating one step further in the prescribed direction with the step-size h:

$$\begin{cases}\hat{x}_j^{i+1}=x_j^{i+1}+h\dfrac{dx_j}{ds}\\ \hat{\lambda}^{i+1}=\lambda^{i+1}+h\dfrac{d\lambda}{ds}\end{cases}\qquad j=1,\cdots,n \tag{5-7}$$

The matlab programme code of predictor is given in Appendix B.

5.5.3 M-file of Corrector

The corrector is the procedure that correct the errors between the predicted approximation $(\hat{x}^{j+1},\hat{\lambda}^{j+1})$ and the next solution (x^{j+1},λ^{j+1}). We choose the Newton-Raphson iterative method to compute the solution, this can be realized by modifying the matpower's

Newton solver to serve as a corrector. Most of work is the modification of the Jacobian matrix and the P&Q mismatch matrix in the Newton solver of Matpower.

From the continuation power flow equations (5-1), we can see that there is one more unknown λ than the conventional power flow equations, therefore, the Jacobian matrix should have one more dimension than the original matrix. We suppose the original Jacobian matrix is:

$$J = [J(x)]$$

and the Jacobian matrix of continuation power flow get one more row and one more column:

$$J = \begin{bmatrix} J(x) & b \\ 2x - 2x(s) & 2\lambda - 2\lambda(s) \end{bmatrix}$$

where b is the changes in real and reactive power load demand and the changes in real power generation.

Table 5-1 illustrates the changes we modify the Jacabian matrix to serve as the corrector, we can note that the modified J has one more dimension than original J.

temp = real(dSbus_dVa(:, [pv; pq]))';
j11 = temp(:, [pv; pq])';
temp = real(dSbus_dVm(:, pq))';
j12 = temp(:, [pv; pq])';
temp = imag(dSbus_dVa(:, [pv; pq]))';
j21 = temp(:, pq)';
temp = imag(dSbus_dVm(:, pq))';
j22 = temp(:, pq)';
 	
J = [j11 j12;
j21 j22;];
	⟹	temp = real(dSbus_dVa(:, [pv; pq]))';
		j11 = temp(:, [pv; pq])';
		temp = real(dSbus_dVm(:, pq))';
		j12 = temp(:, [pv; pq])';
		j13 = - real(b([pv; pq]));
		temp = imag(dSbus_dVa(:, [pv; pq]))';
		j21 = temp(:, pq)';
		temp = imag(dSbus_dVm(:, pq))';
		j22 = temp(:, pq)';
		j23 = - imag(b(pq));
		
		j31 = 2 * (Va([pv;pq]) - va([pv;pq]))';
		j32 = 2 * (Vm(pq) - vm(pq))';
		j33 = 2 * (A - a);
		
		J = [j11 j12 j13;
		j21 j22 j23;
		j31 j32 j33;];

Table 5-1: Modification of Jacabian matrix in Matpower

mis = V .* conj(Ybus * V) - Sbus;
 	
F = [real(mis([pv; pq]));
 	
imag(mis(pq))];
	⟹	mis = V .* conj(Ybus * V) - A * b - Sbus;
		
		Mis = sum((Va([pv;pq]) - va([pv;pq])) .^2)
		+ sum((Vm(pq) - vm(pq)) .^2) + (A - a)^2 - ds^2;
		
		F = [real(mis([pv; pq]));
		imag(mis(pq));
		Mis];

Table 5-2: Modification of P&Q mismatch matrix in Matpower

The P&Q mismatch matrix F is changed from $F=\begin{bmatrix}\hat{P}-P\\ \hat{Q}-Q\end{bmatrix}$ to $F=\begin{bmatrix}\hat{P}-P\\ \hat{Q}-Q\\ \Delta\hat{s}^2-\Delta s^2\end{bmatrix}$

Where the P is the real power of all the PV and PQ buses, and Q is the reactive power of all PQ buses.

Table 5-2 illustrates the changes we modify the original mismatch matrix, we can note that the modified mismatch matrix F has one more row element in the matrix. The matlab program code is presented in Appendix C.

5.2.4 Choice of Step Size

The step size is an important element that affects the computational accuracy. If we choose a small step size in CCPF computation, it is safe to pass the critical point without divergence, but this will take too many continuation steps in the flat part of P-V curve, and make the computation inefficient.

The ECPF developed by Mori and Yamada proposes a new step size control rule, which improves the CCPF method efficiently. It takes large step size in the flat part of P-V curve, and adjust the value of step size to smaller and smaller according to the gradient of P-V curve.

The Q-limit guided CPF method can avoid the trouble of the selection of step size, because the step-size can be efficiently guided by the Q-limit breaking index. We just choose a small step size to secure the accuracy requirement of computation, and then set the value of original step-size as a constant. During the computational process, the step size of some continuation step will be changed by

$$h_q = \lambda_q - \lambda_n$$

where the h_q is the step size, the λ_q is the Q-limit index, the λ_n is the value of the parameter λ of current continuation step. Basically, the value of h_q depends on where the reactive power of generation will reach its limit. After the Q-limit breaking point is passed, the step size h_q will be change back to the initial value, and then go to the next process.

The proposed GCPF uses Q-limit indices to change the step size, so, just pre-set the step size that fit the requirement of accuracy requirement. The step size would be guided feasible by the Q-limit breaking indices.

5.2.5 Q-limit Index Predictor

In order to realize this procedure in Matpower, a new function of m-files is needed. The algorithm of Q-limit index computation can be described by the six steps below:

1). At first, Compute three solutions of conventional continuation power flow.

2). Then, to compute the reactive power Q_G for every generator in every solution.

3). For each generator j, construct a matrix

$$A_j = \begin{bmatrix} Q_{G_{j0}}^2 & Q_{G_{j0}} & 1 \\ Q_{G_{j1}}^2 & Q_{G_{j1}} & 1 \\ Q_{G_{j2}}^2 & Q_{G_{j2}} & 1 \end{bmatrix}$$

and construct a vector

$$\Lambda = [\lambda_0, \lambda_1, \lambda_2]^T$$

$$g_j = [a_j, b_j, c_j]^T$$

4). Solve $A_j g_j = \Lambda$, we can get the coefficient a, b, c of generator j for the equation below:

$$\lambda = a_j Q_{G_j}^2 + b_j Q_{G_j} + c_j$$

5). Calculate the extreme values $q_{G_j} = -\frac{b_j}{2a_j}$ and obtain a list of generator buses that satisfy

$$\text{if } q_{G_j} > 0,\ q_{G_j} \geq Q_{G\max} \quad \text{or}$$

$$\text{if } q_{G_j} < 0,\ q_{G_j} \leq Q_{G\min}$$

These buses in the list take violation to the reactive power resource, so I call this list violation list.

6). To get the Q-limit index, we substitute the $Q_{\min}$ and $Q_{\max}$ values for each generator bus j that is in the violation list into the interpolated polynomials

$$\lambda_{q\min}^j = a_j Q_{j\min}^2 + b_j Q_{j\min} + c_j$$

$$\lambda_{q\max}^j = a_j Q_{j\max}^2 + b_j Q_{j\max} + c_j$$

then we obtain a series of $\lambda^{j}_{q\min}$ and $\lambda^{j}_{q\max}$, In each case, only the smallest value from them will be of interest.

We can represent the procedure (6) as follows:

$$\lambda_q = \min(\lambda^{i}_{q\min}, \lambda^{i}_{q\max})$$

Subject to $\lambda_q > \lambda_0$, we take $\Delta s = \lambda_q - \lambda_0$ as the new step-size to guide the CPF method from the current point (x_0, λ_0) to the new operating point (x_q, λ_q). We then change Δs back to the initial value to carry on the CPF computation until the Q-limit breaking point is reached and then change the bus type from PV to PQ. We then set this Q-limit breaking point as the new starting point (x_0, λ_0) and the whole process is repeated again until the bifurcation point is reached.

The matlab programme code of Q-limit index predictor is given in Appendix D.

5.3 Summary

We create a package of m-files for solving CPF problems in Matpower, It is intended as a simulation tool for researchers and educators that is easy to use. The code is designed to give the best performance possible while keeping the code simple to understand and modify. Some of the program code are presented in appendices.

Chapter 6: Comparison and Analysis of CCPF, ECPF and GCPF

Chapter 6 is the full results of comparisons between CCPF, ECPF and the proposed GCPF method. I have made the comparisons in multiple means, which includes illustration figures and numerical results tables. A performance comparison is made between CCPF, ECPF and GCPF methods in terms of required continuation steps, computational time and accuracy.

6.1 Introduction

In this section, I have made plenty of experiments. I compared the proposed Q-limit guided continuation power flow (GCPF) method with the conventional continuation power flow method (CCPF) and the enhanced continuation power flow (ECPF) in required continuation steps, computation time and accuracy of maximum load parameter. We would know how GCPF method works and how efficient it is.

In the experiments, we consider two kinds of load patterns: one is single load bus variation pattern, which means to pick up one single load bus from the power system to vary, for simplicity we choose the first PQ bus in each data set. Another is multiple load bus variation pattern, which means to pick up more than one load bus to vary, for simplicity we choose all the load buses of one area if the system network is divided into areas and has less than ten load buses, if not we choose the first ten load buses in each data set instead or the first twenty five load buses if the system is very large.

In order to meet the requirement of computing accuracy, we set the basically initial step size to be 0.05 for the CCPF method and GCPF method except at some special GCPF steps when a valid Q-limit index is computed. The starting step size of ECPF method is 0.15, and then reduced to 0.10, finally the step size 0.05 is employed that is to secure the same accuracy requirement with other methods.

6.2 Comparison and Analysis of CCPF, ECPF and GCPF

These meaningful comparisons are conducted on the following 5 power system case studies:

- IEEE 9: 3 generator buses and 3 load buses;
- IEEE 30: 6 generator buses and 20 load buses;
- IEEE 118: 54 generator buses and 99 load buses;
- National Grid (NG) UK 61 bus system [92];
- National Grid (NG) UK 953 bus system.

6.2.1 IEEE 9 Bus System Case

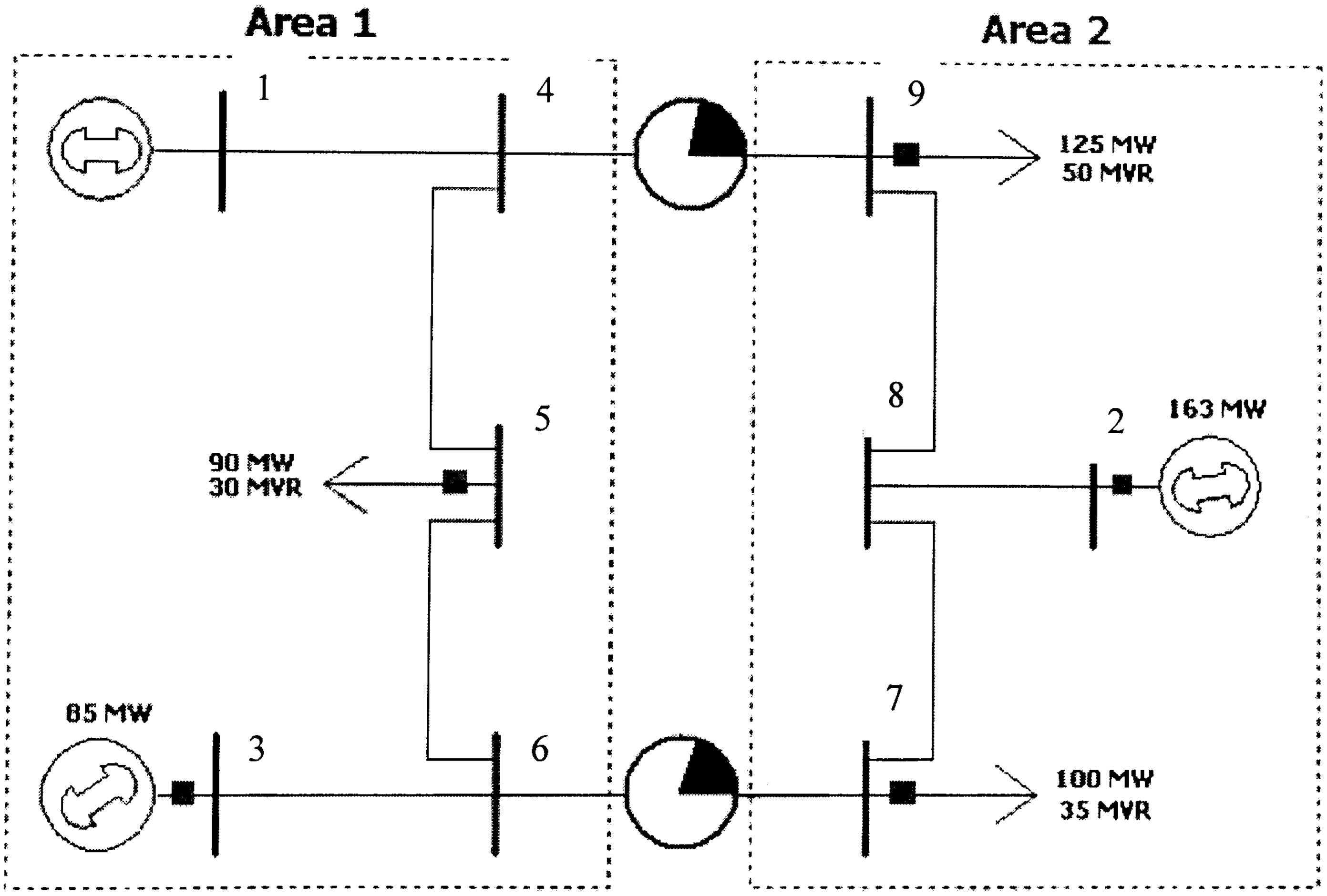


Figure 6-1: Illustration of IEEE 9-bus system

IEEE 9 bus system is a quite simple power system, just has 3 generators and 3 loads. The system has been divided into two areas. The summary of the system is illustrated in Figure 6-1 and Table 6-1.

How many		How much	P (MW)	Q (MVAR)
Buses	9	Total Gen Capacity	820.0	-900.0 to +900.0
Generators	3	On-line Capacity	820.0	-900.0 to +900.0
Committed Gens	3	Generation (current)	320.0	34.9
Loads	3	Load	315.0	115.0
Branches	9	Losses (I^2*Z)	4.95	51.31
Transformers	0	Branch Charging	-	131.4
Areas	2	Shunt	0.0	0.0
Inter-ties	0	Total Inter-tie Flow	0.0	0.0

Table 6-1: The summary of IEEE 9 bus system

The experiments are implemented in single load bus variation and multiple load buses variation respectively.

Single Load Bus Variation Case

There are 3 load buses totally in IEEE 9 bus system, load bus 5 in area one is chosen to vary for this case, both the real power and reactive power of bus 5 increase at equivalent rate. In order to compare easily, the P-V curve of load bus 5 is plotted using CCPF, ECPF and GCPF methods respectively, and the numerical results are also summarized in tables.

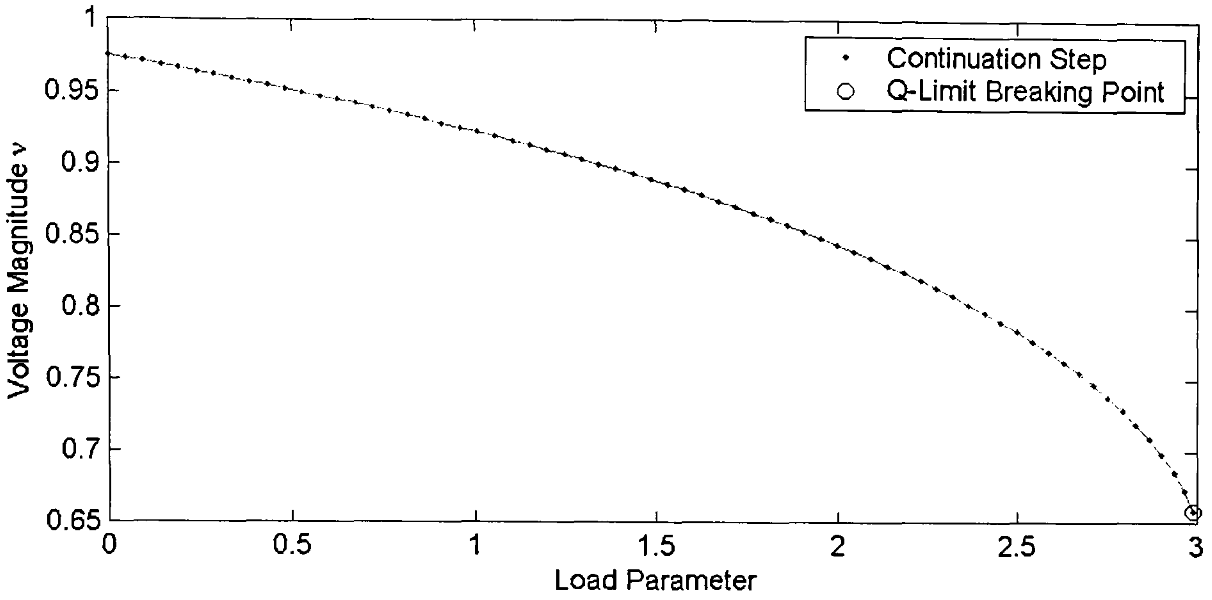


Figure 6-2-a: Varying single load bus of IEEE 9 bus system using CCPF

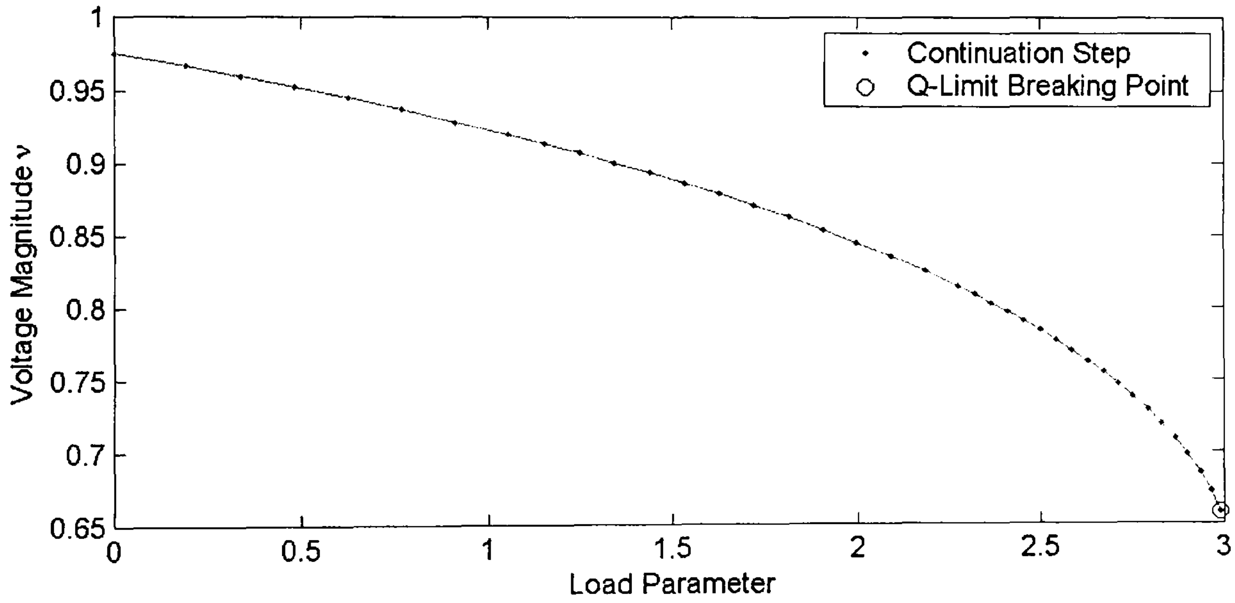


Figure 6-2-b: Varying single load bus of IEEE 9 bus system using ECPF

Figure 6-2-a, 6-2-b and 6-2-c illustrate the curve of voltage magnitude of load bus 5 versus load parameter λ. Figure 6-2-a was obtained using CCPF method, Figure 6-2-b was obtained using ECPF method and Figure 6-2-c was obtained using the novel GCPF method. In Figures 6-2-a, 6-2-b and 6-2-c, we use dots to denote continuation steps and the symbol "o" is used to denote the steps where the Q-limit breaking point occurs.

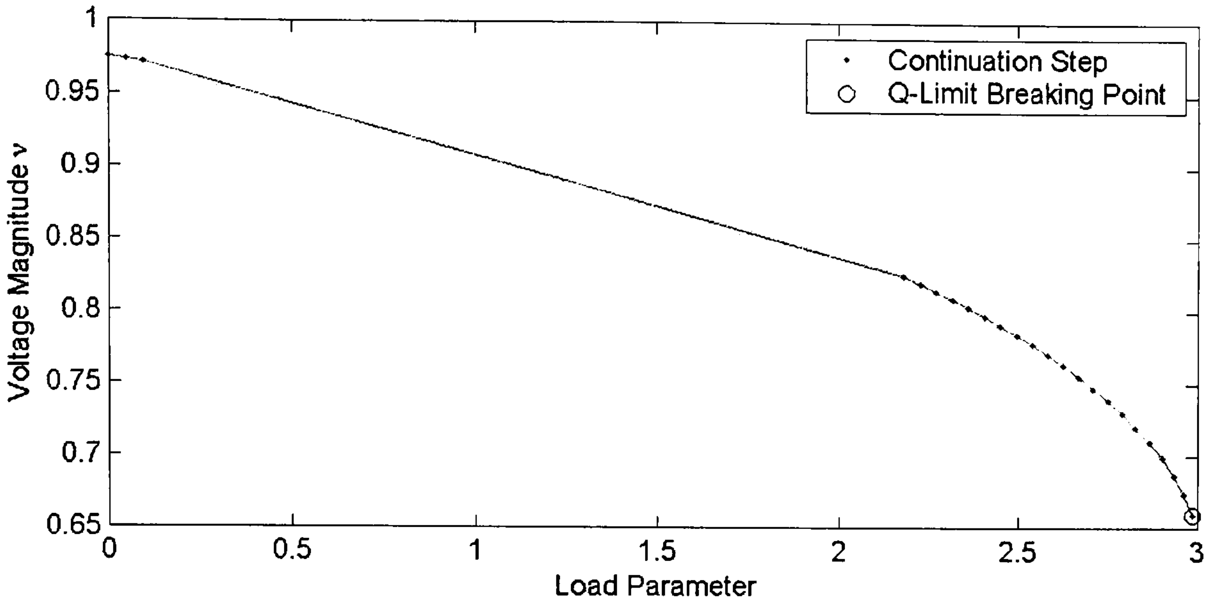


Figure 6-2-c: Varying single load bus of IEEE 9 bus system using GCPF

	Bus Type Changes	Continuation Steps	Computation time(s)	Load Parameter
CCPF	1	67	0.5684	2.9909
ECPF	1	39	0.3780	2.9909
GCPF	1	24	0.2406	2.9890

Table 6-2: Results summary of varying single load bus of IEEE 9 Bus System

Discussion

From Figure 6-2-a, the P-V curve of load bus 5 is plotted using CCPF method, which takes too many continuation steps to pass the bifurcation point due to the small step size chosen in the whole process. In this case, there is only one bus type changed nearby the bifurcation point, and this bus was the original slack bus, we define the PV bus next to the original slack bus in the data set to be the new slack bus in our algorithm after the slack generator bus is changed to PQ bus type. From the numerical result table 6-2, we observe that CCPF method is reliable and stable but not efficient; it takes 67 continuation steps and spends 0.5684 seconds to pass the bifurcation point.

From Figure 6-2-b, the P-V curve of load bus 5 is plotted using ECPF method, the rule based step size control mechanism does improve the CCPF method. From the P-V curve, it is clearing apparent that different step size is implemented between the continuation steps. The large step size is used in the flat part of the P-V curve, and the step size is reduced with the increasing of the curvature of the P-V curve. The step size would be reduced further with larger curvature of the curve, finally, changed back to the same step size used in CCPF method in order to secure the requirement of accuracy. ECPF works more efficient than CCPF, which can be found from the numerical results Table 6-2, ECPF takes 39 continuation steps and spends 0.3780 seconds passing the critical point.

From Figure 6-2-c, the P-V curve of load bus 5 is plotted using GCPF method, from the overview of the curve, the continuation steps are not continuously, there are three continuation steps at the beginning of the curve that is for the data collection of Q-limit index predictor. When the Q-limit index is computed, the step size will be enlarged, and then the current continuation step will be guided to the Q-limit breaking point in the fast way possible. It is easy to observe this procedure from the curve of Figure 6-2-c, there is a long gap on the curve, which is caused by the Q-limit index predictor. After the continuation step is guided by the enlarged step size, the step size will be changed back to the initial value to carry on the CPF computation until the Q-limit breaking point is passed, and then repeat the procedure of Q-limit predictor to compute another three solutions for the next computation of Q-limit index prediction. In this case, GCPF just performs once Q-limit index predictor, because the bifurcation point is passed nearby the Q-limit breaking point. From Table 6-2, the proposed GCPF method works more efficient than CCPF and GCPF, which takes 24 continuation steps and spends 0.2406 seconds passing the critical point.

In this single load bus variation experiment on IEEE 9 bus system, the data condition is not very good to implement GCPF method, because GCPF is seriously depend on the distribution of Q-limit breaking points. In this experiment, there is only one generation bus reaches its Q-limit and it is quite far from the base solution, which causes the Lagrange polynomial interpolation method is not able to predict the Q-limit index accurately, that's why GCPF takes many continuation steps before the Q-limit breaking point in Figure 6-2-c. In this case, we expected less than 10 continuation steps including the first three data collection solutions. In general, GCPF is a quite efficient method to compute bifurcation point, which takes 43 and 15 continuation steps less than CCPF and ECPF respectively, and saves 0.3278 and 0.1374 seconds comparing to the CCPF and ECPF respectively.

Multiple Load Buses Variation Case

There are 3 load buses totally in IEEE 9 bus system, load buses 7 and 9 in area two are chosen to vary for this multiple case, and both the real power and reactive power of load bus 7 and 9 increases in equivalently rate. In order to compare easily, the P-V curve of load bus 5 is plotted using the CCPF, ECPF and GCPF methods respectively, and the numerical results are also summarized in tables.

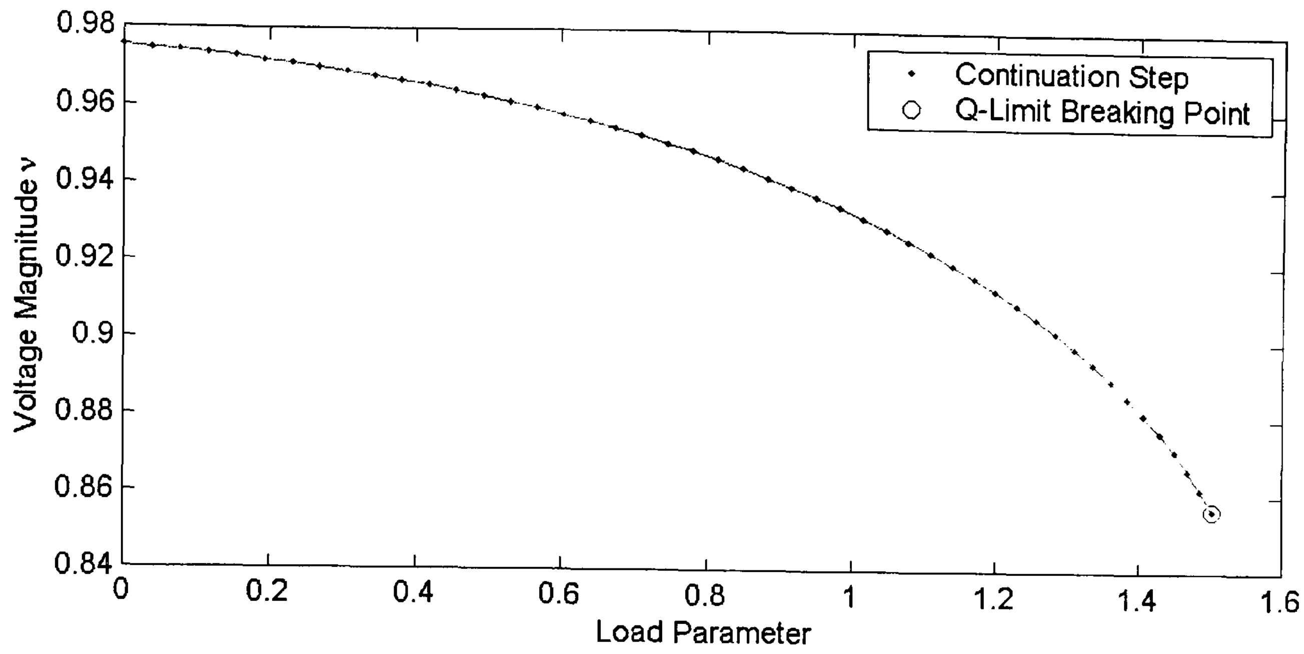


Figure 6-3-a: Varying multiple load buses of IEEE 9 bus system using CCPF

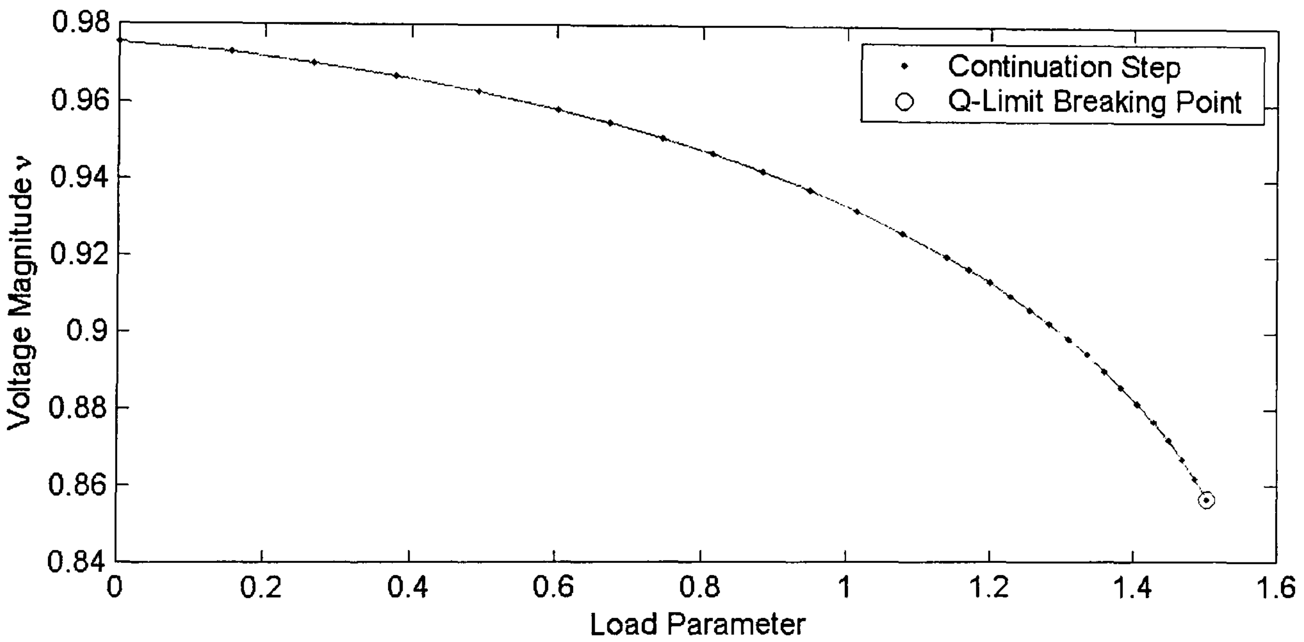


Figure 6-3-b: Varying multiple load buses of IEEE 9 bus system using ECPF

Figures 6-3-a, 6-3-b and 6-3-c illustrate the curve of voltage magnitude of load bus 5 versus load parameter λ. Figure 6-3-a was obtained using CCPF method, Figure 6-3-b was obtained using ECPF method and Figure 6-3-c was obtained using the novel GCPF method. In Figures 6-3-a, 6-3-b and 6-3-c, we use dots to denote continuation steps and the symbol “o” is used to denote the step where the Q-limit breaking point occurs.

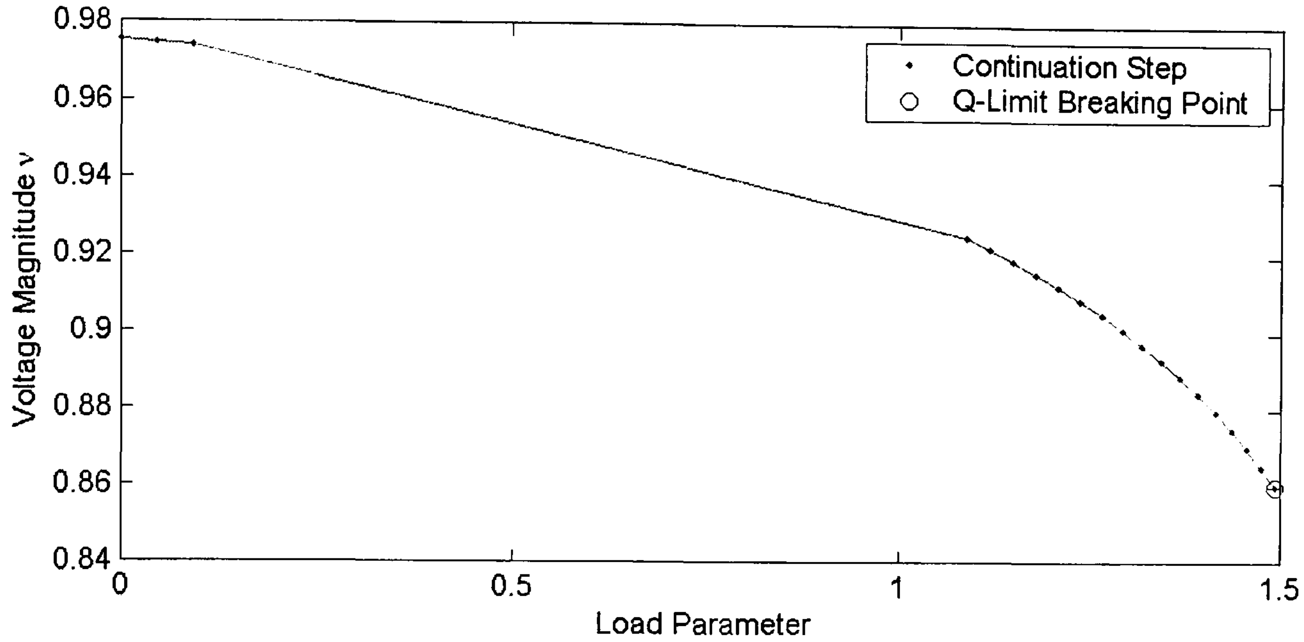


Figure 6-3-c: Varying multiple load buses of IEEE 9 bus system using GCPF

	Bus Type Changes	Continuation Steps	Computation time(s)	Load Parameter
CCPF	1	48	0.2420	1.5034
ECPF	1	29	0.1720	1.5034
GCPF	1	20	0.1325	1.4928

Table 6-3: Results summary of varying multiple load buses of IEEE 9 Bus System

Discussion

From Figure 6-3-a, the P-V curve of load bus 5 is plotted using CCPF method, the continuation steps distribute serried on the P-V curve that means CCPF still takes a large number of continuation steps for the computation of bifurcation point. There is one Q-limit breaking point occurs during the whole computational steps, which was the initial slack bus, the next PV bus is changed to be the new slack bus. CCPF method is a reliable and stable but not efficient compared with ECPF and GCPF methods. CCPF takes 48 continuation steps and spends 0.2420 seconds passing the bifurcation point from Table 6-3.

From Figure 6-3-b, the P-V curve of load bus 5 is plotted using ECPF method. ECPF uses large step size at the start of the computation, and reduces the step size with the gradient of curve increasing that is illustrated in the Figure 6-3-b clearly. The continuation steps locate sparsely in the "flat" part of the curve, then the location of steps are becoming more and more dense closing to the critical point at end of the curve, which is affected by the rule based step size control mechanism applied in ECPF. From the numerical results table, ECPF method takes 29 continuation steps and spends 0.1720 seconds passing the critical point, which proves that ECPF is more efficient method than CCPF, and did improve the CCPF.

From the Figure 6-3-c, the P-V curve of load bus 5 is plotted using the novel GCPF method. The data set of IEEE 9 bus system did not provide good condition to implement GCPF, the reason has been discussed in the single load bus variation case, because the efficiency of GCPF strongly depends on the distribution of Q-limit breaking point. In this multiple load bus variation case, there is only one reactive power output of generator that has reached its limit, so the step size is efficiently adjusted only once during the whole computation process, the Lagrange polynomial interpolation can not predict very accurately under this condition. From Table 6-3, we see GCPF takes 20 continuation steps and spends 0.1325 seconds completing the computation. GCPF is still faster than CCPF and ECPF, but not as we expected less than 10 steps.

In this multiple load bus variation case, we get the similar results with the single load bus variation case. From the numerical result in Table 6-3, we see that GCPF takes 28 and 9 continuation steps less than CCPF and ECPF methods respectively, and saves 0.1095 and 0.0395 seconds respectively. The proposed GCPF is quite efficient than CCPF, but is comparable to the ECPF.

The IEEE 9 bus system provides a special condition that is only one Q-limit breaking point occurs closing to the bifurcation point and the Q-limit breaking point is quite far from the base case solution. So the Q-limit index predictor procedure did not predict very accurately, which causes GCPF method takes many continuation steps before Q-limit breaking point in the Figures 6-2-c and 6-3-c. So, it comes to a conclusion that implementation of GCPF is relying on the location and quantity of Q-limit breaking point and the novel GCPF is much more efficient than the conventional CPF method.

6.2.2 IEEE 30 Bus System Case

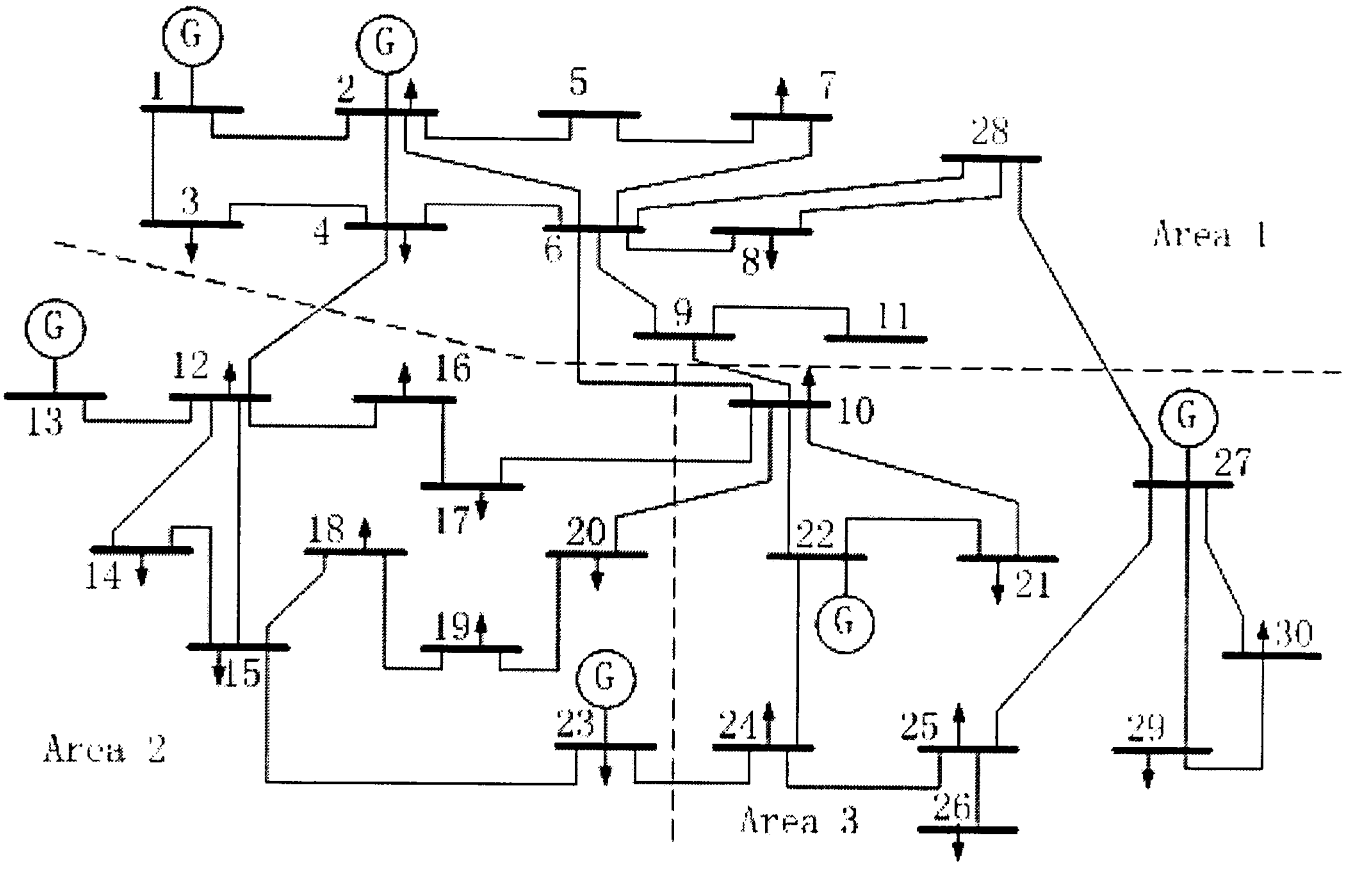


Figure 6-4: Illustration of IEEE 30 Bus System

How many		How much	P (MW)	Q (MVAR)
Buses	30	Total Gen Capacity	335.0	-95.0 to +405.9
Generators	6	On-line Capacity	335.0	-95.0 to +405.9
Committed Gens	6	Generation (current)	191.6	100.4
Loads	20	Load	189.2	107.2
Branches	41	Losses (I^2*Z)	2.44	8.99
Transformers	0	Branch Charging	-	15.6
Areas	3	Shunt	0.0	0.2
Inter-ties	7	Total Inter-tie Flow	66.4	54.2

Table 6-4: The summary of IEEE 30 bus system

IEEE 30 bus system is a well tested case, has 6 generators, 20 load buses and 41 branches. The network of IEEE 30 has been divided into three areas as illustrated in Figure 6-4 [93].

Single Load Bus Variation Case

We choose the load bus 21 in area three to vary for this case, both the real power and reactive power of bus 21 increase at equivalent rate. In order to compare easily, the P-V curve of load bus 3 is plotted using CCPF, ECPF and GCPF methods respectively, and the numerical results are summarized in tables.

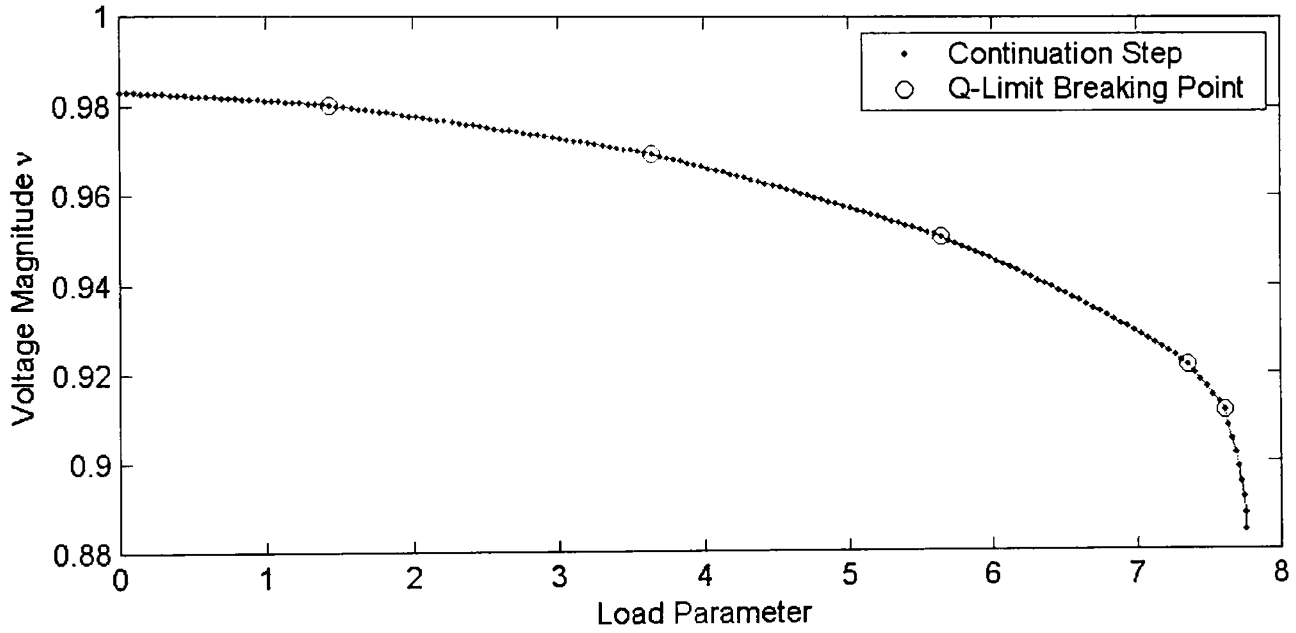


Figure 6-5-a: Varying single load bus of IEEE 30 bus system using CCPF

Figure 6-5-a, 6-5-b and 6-5-c illustrate the curve of voltage magnitude of load bus 3 versus load parameter λ. Figure 6-5-a was obtained using the CCPF method, Figure 6-5-b was obtained using ECPF method and Figure 6-5-c was obtained using the novel GCPF method. In Figures 6-5-a, 6-5-b and 6-5-c, we use dots to denote continuation steps and the symbol "o" is used to denote the step where the Q-limit breaking point occurs.

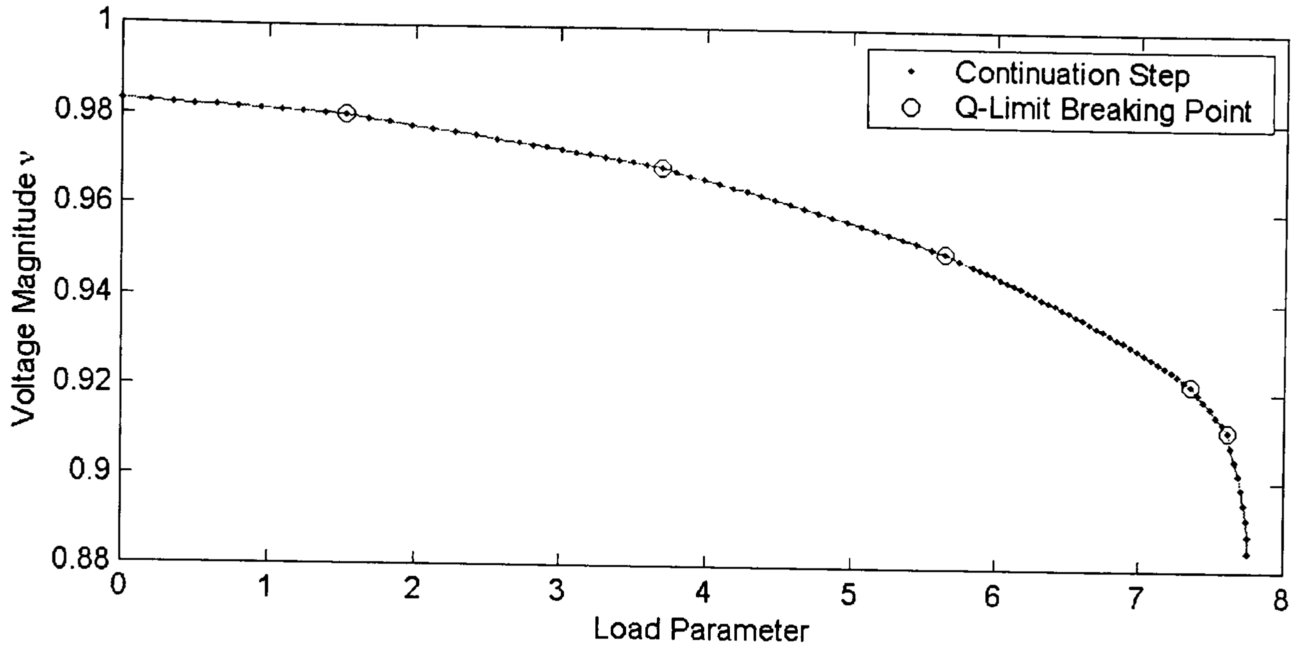


Figure 6-5-b: Varying single load bus of IEEE 30 bus system using ECPF

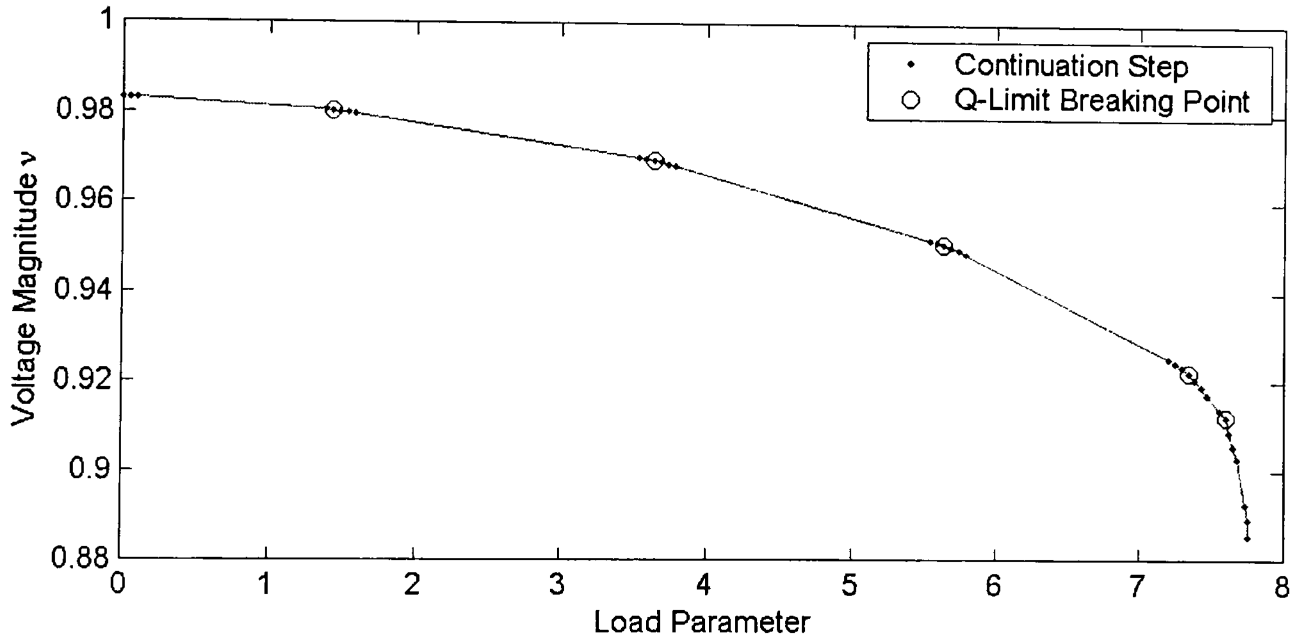


Figure 6-5-c: Varying single load bus of IEEE 30 bus system using GCPF

	Bus Type Changes	Continuation Steps	Computation time(s)	Load Parameter
CCPF	5	166	0.8280	7.7584
ECPF	5	97	0.5470	7.7584
GCPF	5	35	0.4060	7.7579

Table 6-5: Results summary of varying single load bus of IEEE 30 Bus System .

Discussion

From Figure 6-5-a, the P-V curve of load bus 3 is plotted using CCPF method. From the viewing of the P-V curve, CCPF takes too many continuation steps to pass the bifurcation point; from the numerical results Table 6-5, the CCPF method takes 166 continuation steps totally and spends 0.8280 seconds finishing the critical point calculation. There are five Q-limit breaking points during the whole computing process, and these five points distribute equably on the P-V curve, which looks like a good condition to implement the proposed GCPF method.

From Figure 6-5-b, the P-V curve of load bus 3 is plotted using ECPF method. From the view of the curve, it is quite clear how the rule based step size control mechanism works. At the start of the curve, the distance between the continuation steps is large, the distance becomes smaller in the middle of the curve, and the distribution of the continuation steps become very dense in the rest of the curve due to the step size being changed back to the initial value. From the numerical results Table 6-5, ECPF takes 97 continuation steps, and spends 0.5470 seconds reaching the critical point. These results prove that ECPF is an efficient method, and improved the CCPF distinctly.

From Figure 6-5-c, the P-V curve of load bus 3 is plotted using our proposed GCPF method. These results are what the GCPF method expected to obtain, GCPF method reduces the number of continuation steps required significantly, therefore, the GCPF method saves substantial computation time. From the numerical results in Table 6-5, it is clear to see that the proposed GCPF method requires much less continuation steps. In this case GCPF takes 35 continuation steps, spends 0.4060 seconds finishing the computation. The maximal load parameter calculated by the GCPF method has the same accuracy with that calculated using CCPF and ECPF. From Figure 6-5-c, we can observe how the GCPF method works, the curve shows us there are five symbols "o", which indicates that five generator buses reached their Q-limit and that all the bus type of these buses have been changed from PV to PQ. If we analyze this curve in detail, it can be seen that initially we compute three solutions by CPF to collecting the data for predicting Q-limit index, then the index changes the step size. The new step size guides the current solution directly to the next continuation step with enlarged step size that is very close to the Q-limit breaking point. After this solution, we change the step size back to the initial value and carry on the CPF procedure until the Q-limit breaking point is reached or the bifurcation point passed, if the former occurs then the bus type of the Q-limit breaking generator would be changed to PQ type. The process is repeated for the computation of the next three continuation steps to collect the data for the next Q-limit breaking index predictions. When there is no more violation of the Q-limits breaking in the final part of the curve, the final index value is not the predicted Q-limit breaking index, but the extreme value instead.

In this single load bus variation case, our GCPF works very well. The IEEE 30 bus system data provides a good condition to implement the GCPF method. There are five Q-limit breaking points occur during the whole process, and locate along the P-V curve equably. In this case, the Lagrange polynomial interpolation predicts the Q-limit indices quite accurately. GCPF just takes a few continuation steps before each Q-limit breaking point, which save a large number of computational steps. In this single load bus variation case, GCPF method presents an excellent performance, the computation speed is much more faster than CCPF and ECPF, GCPF takes 131 and 62 steps less than CCPF and ECPF respectively, and saves 0.4220 and 0.1410 seconds comparing to CCPF and ECPF respectively.

Multiple Load Bus Variation Case

The IEEE 30 bus system has been divided into three areas as illustrated in Figure 6-4 [93]. In this multiple load bus variation case, we choose the variation load buses by area, there are seven load buses in area 3, they are buses 10, 21, 24, 25, 26, 29 and 30, at which both P and Q are varied at equivalent rate. In order to compare easily, the P-V curve of load bus 3 is plotted using CCPF, ECPF and GCPF methods respectively, and the numerical results are summarized in tables.

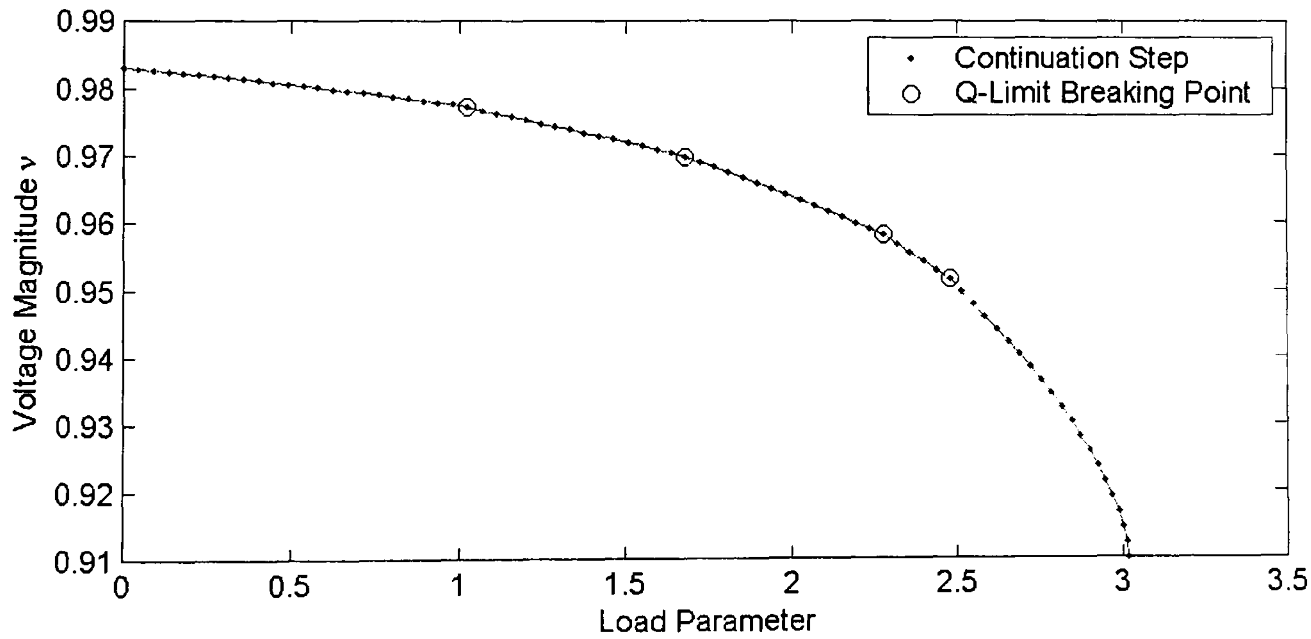


Figure 6-6-a: Varying multiple load buses of IEEE 30 bus system using CCPF

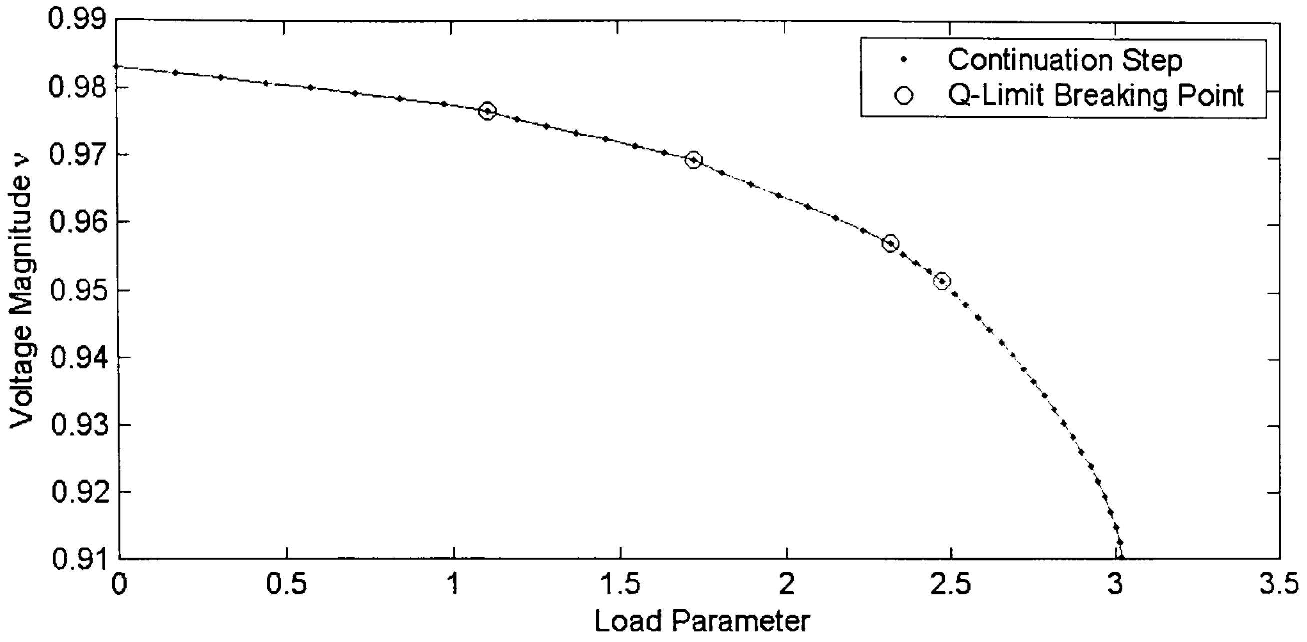


Figure 6-6-b: Varying multiple load buses of IEEE 30 bus system using ECPF

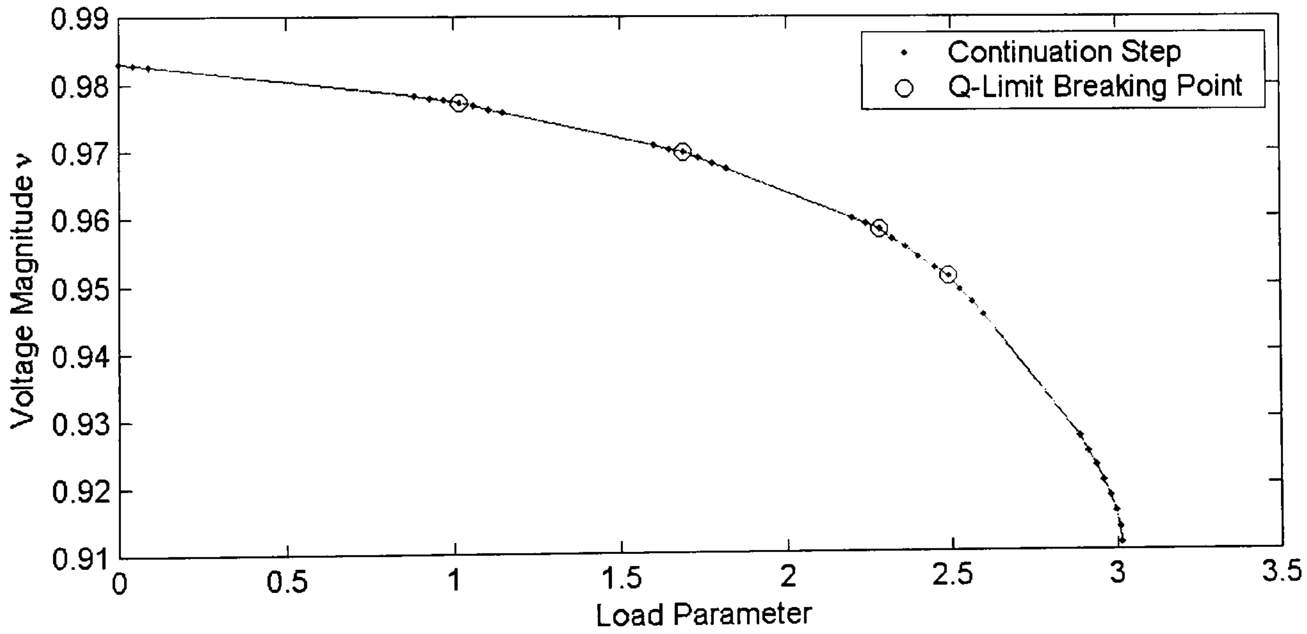


Figure 6-6-c: Varying multiple load buses of IEEE 30 bus system using GCPF

Figure 6-6-a, 6-6-b and 6-6-c illustrate the curve of voltage magnitude of load bus 3 versus load parameter λ. Figure 6-6-a was obtained using the CCPF method, Figure 6-6-b was obtained using ECPF method and Figure 6-6-c was obtained using the novel GCPF method. In Figures 6-6-a, 6-6-b and 6-6-c, we use dots to denote continuation steps and the symbol "o" is used to denote the step where the Q-limit breaking point occurs.

	Bus Type Changes	Continuation Steps	Computation time(s)	Load Parameter
CCPF	4	78	0.6720	3.0166
ECPF	4	47	0.4840	3.0165
GCPF	4	35	0.3750	3.0149

Table 6-6: Results summary of varying multiple load buses of IEEE 30 Bus System

Discussion

From Figures 6-6-a, 6-6-b and 6-6-c, the CCPF and ECPF methods take a large number of continuation steps to reach the bifurcation point, but the GCPF method reduces the number of steps required significantly. Therefore, the novel GCPF method saves substantial computational time. In this case the GCPF method gives another optimal results and performance, and it is quite clear how GCPF works from the curve of Figure 6-6-c.

From Figure 6-6-c, the P-V curve illustrates the optimal algorithm procedure, and presents optimal results. From the view of the curve, the first three continuation solutions are the procedure of data collection of Q-limit index prediction. When the Q-limit index is available, the continuation step skips to the next step with the enlarged step size that is changed by the Q-limit index; this process ignores a large number of unnecessary continuation steps. The Q-limit index prediction procedure would repeat after the predicted Q-limit breaking point reached or bifurcation point passed. In this multiple load bus variation case, we'd like to emphasize that the last Q-limit index is not the prediction of Q-limit breaking point, but the extreme value of quadratic P-Q curve predicted by Lagrange polynomial interpolation formula, which could be observed at the end of the P-V curve of Figure 6-6-c, there is no more Q-limit breaking point occurring in the last series of continuation solutions until the bifurcation point is passed.

In this multiple load bus variation case, we could draw the similar conclusions to the single load bus case. Our proposed GCPF is an excellent method, which takes 43 and 12 steps less than CCPF and ECPF respectively, and saves 0.1880 and 0.1090 seconds in the computation time respectively. The data set of IEEE 30 bus system provides a very good condition to implement GCPF.

6.2.3 IEEE 118 Bus System Case

The IEEE 118 bus system is a complicated system, which has 54 generation buses and 99 load buses. The difference from the above two cases is that this case has 9 transformers and the buses didn't be divided into areas, all the buses are in one area in the data set. So, the variation load buses may not be chosen by area in this case, we choose the first ten load buses to be the multiple variation load bus from the base data set.

Single Load Bus Variation Case

IEEE 118 bus system hasn't been divided into areas, so we pick up the first load bus as the variation bus; the first PQ bus is bus 3 in the IEEE 118 data set, the both P and Q at bus 3 to be varied at equivalent rate. In order to compare easily, the P-V curve of load bus 3 is plotted by using CCPF, ECPF and GCPF methods respectively, and the numerical results are summarized in tables.

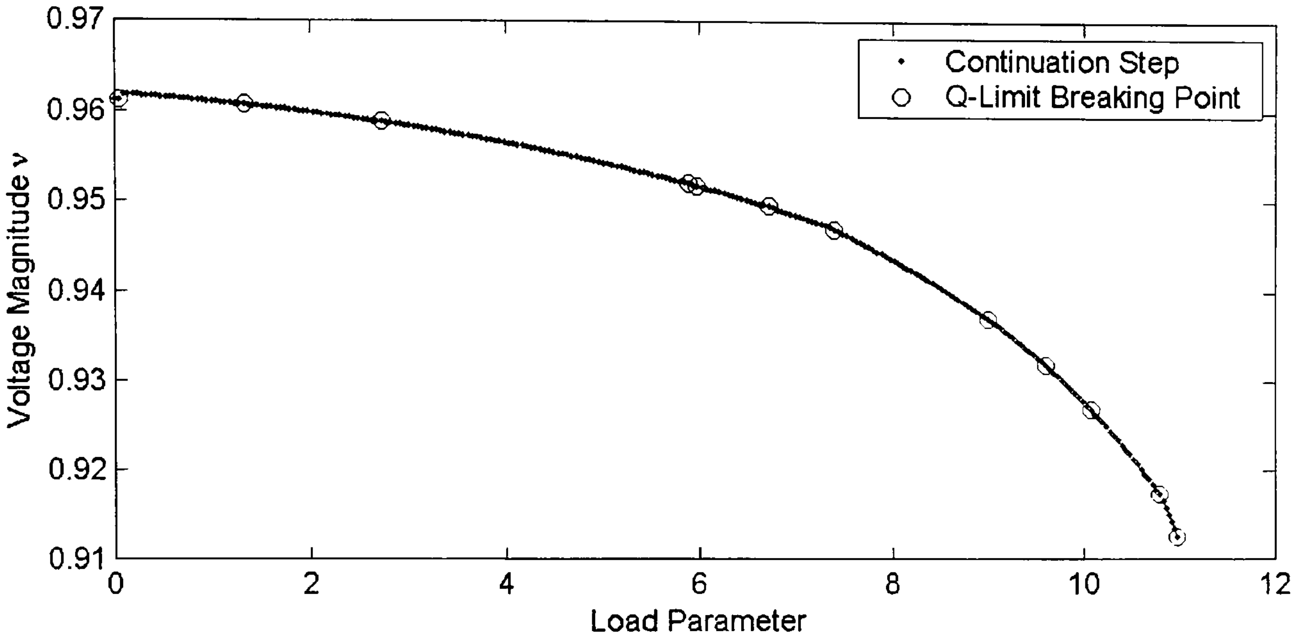


Figure 6-7-a: Varying single load bus of IEEE 118 bus system using CCPF

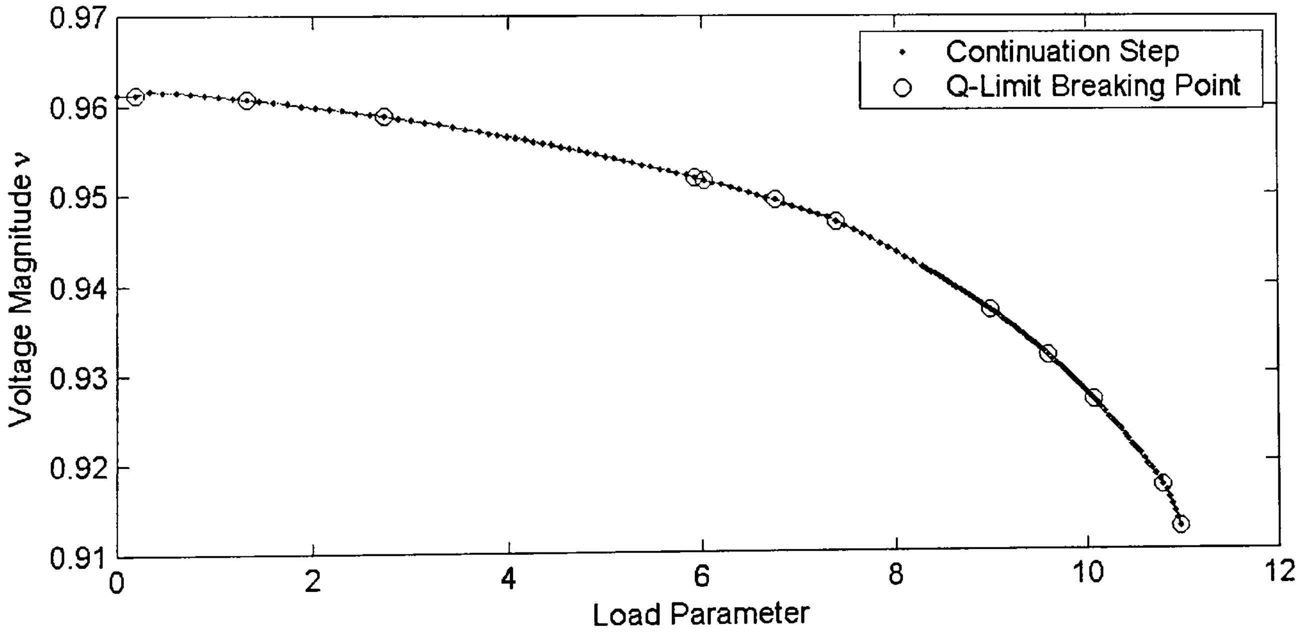


Figure 6-7-b: Varying single load bus of IEEE 118 bus system using ECPF

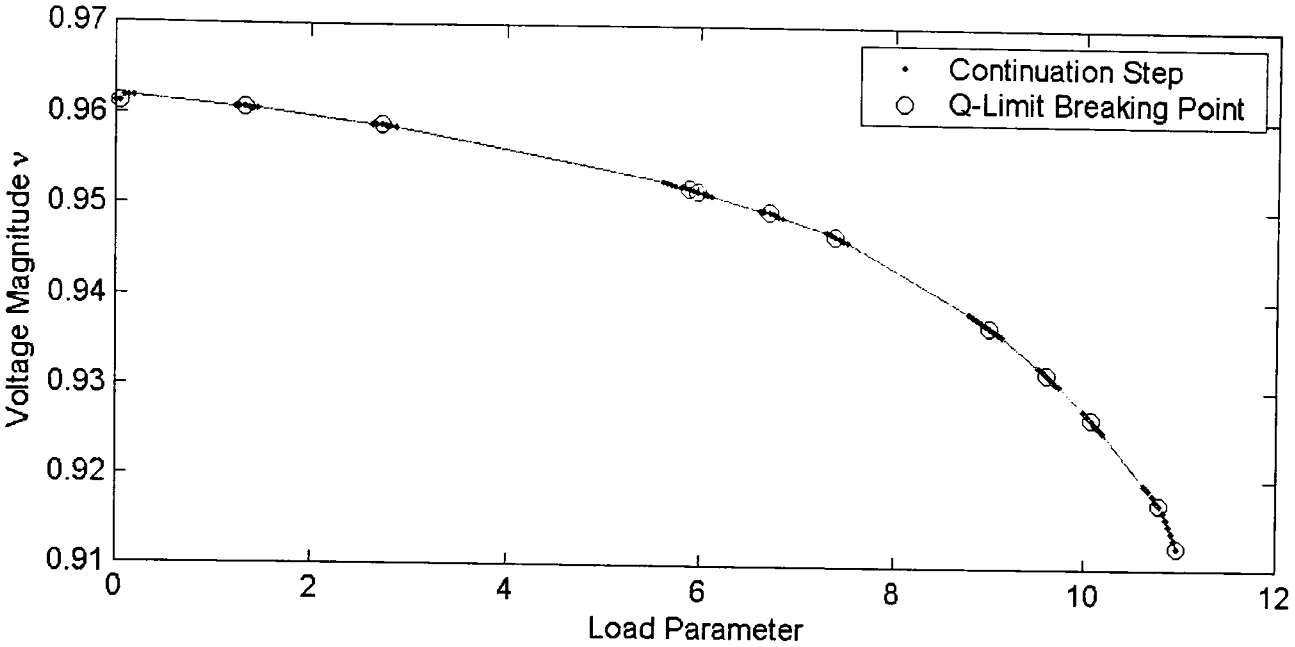


Figure 6-7-c: Varying single load bus of IEEE 118 bus system using GCPF

Figure 6-7-a, 6-7-b and 6-7-c illustrate the curve of voltage magnitude of load bus 38 versus load parameter λ. Figure 6-7-a was obtained using the CCPF method, Figure 6-7-b was obtained using ECPF method and Figure 6-7-c was obtained using the novel GCPF method. In Figures 6-7-a, 6-7-b and 6-7-c, we use dots to denote continuation steps and the symbol "o" is used to denote the step where the Q-limit breaking point occurs.

	Bus Type Changes	Continuation Steps	Computation time(s)	Load Parameter
CCPF	17	243	3.2184	10.9720
ECPF	17	141	2.2575	10.9725
GCPF	17	71	1.5620	10.9620

Table 6-7: Results summary of varying single load bus of IEEE 118 Bus System

Discussion

From the overview of Figures 6-7-a, 6-7-b and 6-7-c, we could draw the similar conclusions to the case of IEEE 30 bus system. The proposed GCPF method is always faster than other methods, GCPF takes 172 and 70 continuation steps less than CCPF and ECPF respectively, and saves 1.6564 and 0.6955 seconds respectively. In this case, there are 17 generator buses that reach their Q-limit, there should be 17 Q-limit breaking points on the P-V curve, but there are 12 Q-limit breaking points on the curve actually, which is because some generator buses break their Q-limit in a same continuation step during the computation process. We can also note that at the start of the P-V curve of Figure 6-7-c, there is a Q-limit breaking point that occurs in the procedure of data collection of Q-limit index prediction, in this case, we have to compute another three solutions to predict Q-limit index after passing the Q-limit breaking point. This situation also happened in the rest of computation process, we can see that there is another Q-limit breaking point that occurs in the procedure of data collection in the fourth departed series of continuation steps on the curve, so we restart to compute another three continuation steps in our GCPF algorithm.

Multiple Load Buses Variation Case

In this case, we choose the variation load buses by the sequence of PQ bus in the IEEE 118 system data set due to the fact that the system hasn't been divided into areas, the first ten PQ buses are picked up to vary, they are bus 2, 3, 5, 7, 9, 11, 13, 14, 16 and 17, at which both P and Q are varied at equivalently rate. In order to compare easily, the P-V curve of load bus 38 is plotted by using CCPF, ECPF and GCPF methods respectively, and the numerical results are summarized in tables.

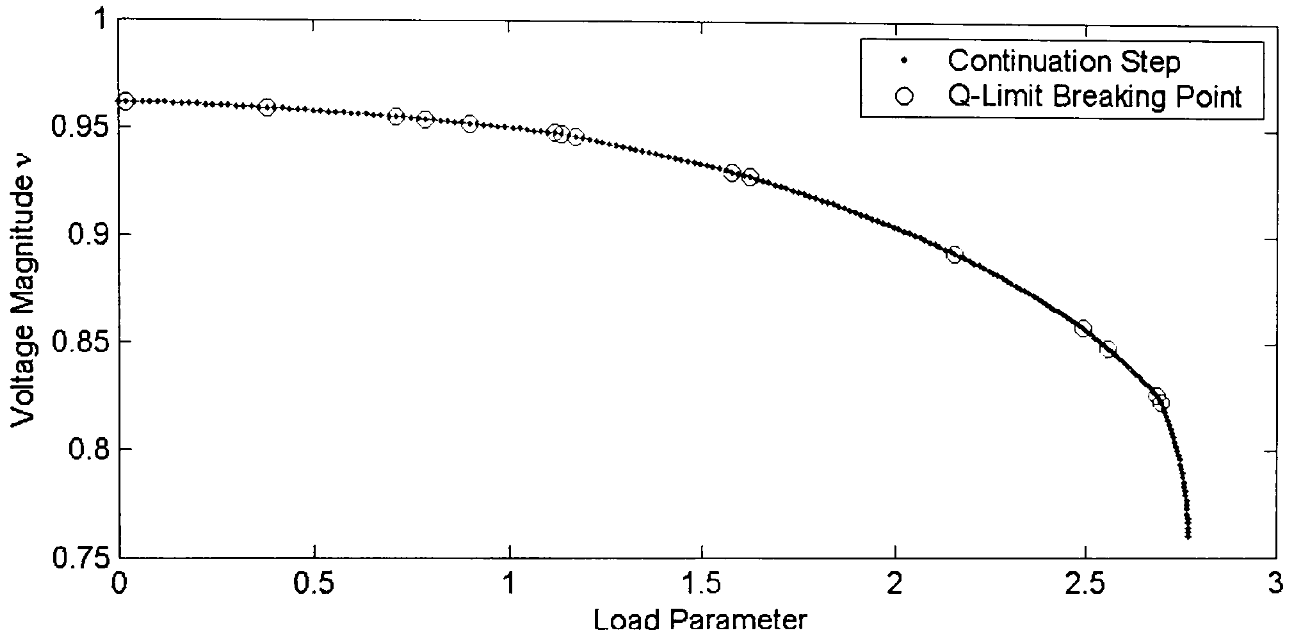


Figure 6-8-a: Varying multiple load buses of IEEE 118 bus system using CCPF

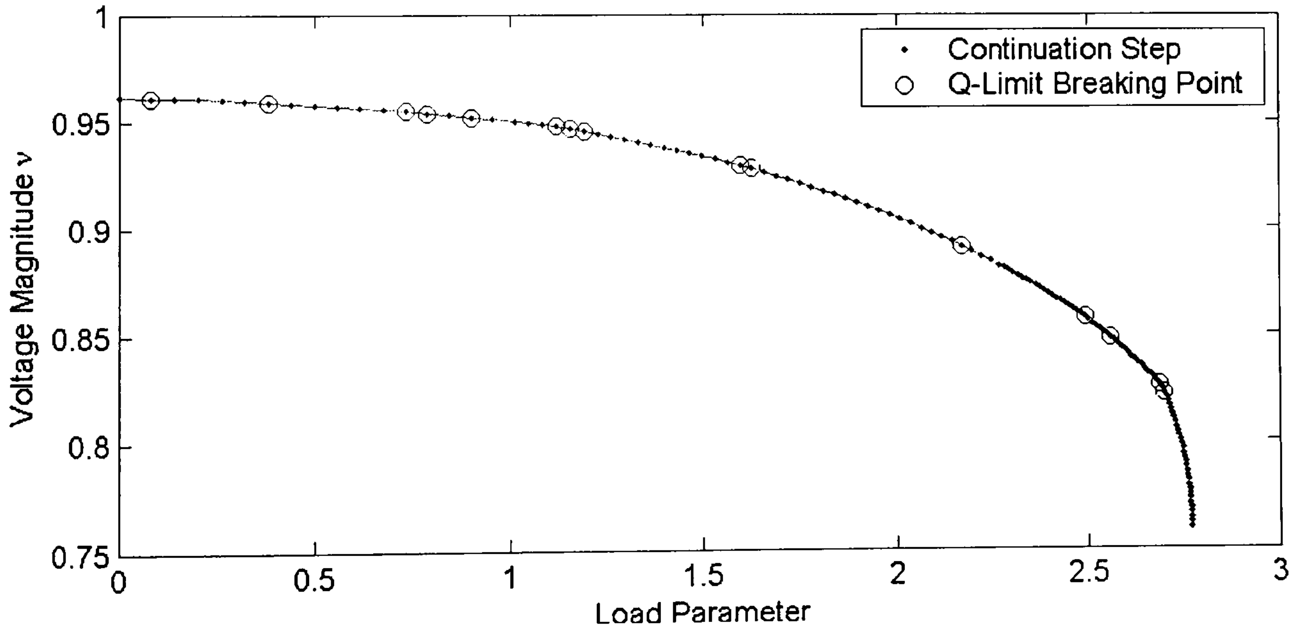


Figure 6-8-b: Varying multiple load buses of IEEE 118 bus system using ECPF

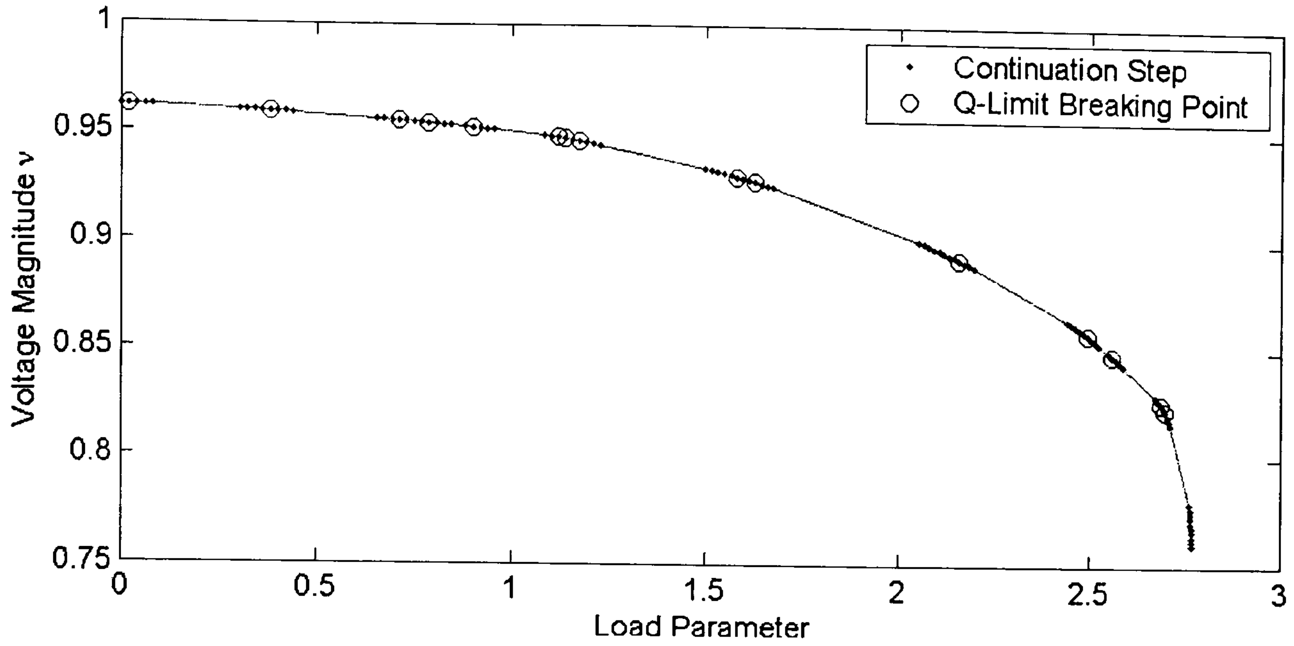


Figure 6-8-c: Varying multiple load buses of IEEE 118 bus system using GCPF

Figure 6-8-a, 6-8-b and 6-8-c illustrate the curve of voltage magnitude of load bus 38 versus load parameter λ. Figure 6-8-a was obtained using CCPF method, Figure 6-8-b was obtained using ECPF method and Figure 6-8-c was obtained using the novel GCPF method. In Figures 6-8-a, 6-8-b and 6-8-c, we use dots to denote continuation steps and the symbol "o" is used to denote the step where the Q-limit breaking point occurs.

	Bus Type Changes	Continuation Steps	Computation time(s)	Load Parameter
CCPF	20	206	2.8280	2.7684
ECPF	20	132	2.2196	2.7684
GCPF	20	92	1.6720	2.7669

Table 6-8: Results summary of varying multiple load buses of IEEE 118 Bus System

Discussion

From the viewing of Figure 6-8-a, 6-8-b and 6-8-c, the ECPF and GCPF methods both reduce the required continuation steps, and improve the CCPF method efficiently. From the numerical results in Table 6-8, the proposed GCPF method takes 114 and 40 steps less than CCPF and ECPF method respectively, saves 1.1560 and 0.5476 seconds in computation time respectively, which is the optimal results we expected to obtain. The data of IEEE 118 bus system provides a good condition to implement the novel GCPF method. In this multiple load buses variation case, we draw the similar conclusions to the single load bus variation case, but there are more Q-limit breaking points that occur in the data collection procedure of Q-limit breaking index prediction. There are 20 generators that reach their Q-limit, but the figures only illustrate 15 Q-limit breaking points on the curve, because some Q-limit breaking points occurred in the same continuation steps.

The IEEE 118 bus system case experiment further proves the performance of our proposed GCPF method that is much better than CCPF, and is more efficient than ECPF. All of these methods compute the bifurcation point with the same requirement of accuracy.

6.2.4 NG 61 Bus System Case

The data was supplied to us by the National Grid Company (NGC), which owns and operates the high voltage electricity transmission system in England and Wales. This NG 61 power system network is on the South West Peninsula, it is one of the parts of the NGC system most prone to voltage collapse. The network is shown in Appendix A. It comprises the 400KV system extending from Dungeness in Kent, and Melksham in Wiltshire, to Indian Queens in Cornwall.

Either of two contingencies shown in Appendix A would cause a large change in the general topology of the network, and leave the Cornwall load electrically more remote from sources of generation. The "South of Hinkley" double circuit outage (SOH) would leave Fawley the remaining major source, supplying load through the south coast 400KV double circuit route. Alternatively, the "East of Mannington" double circuit outage (EOM) would leave the load supplied from Hinkley Point. Exeter is a critical bussing point for reactive power support to the Cornwall network.

Indian Queens, being the most remote point in the network is most prone to voltage collapse under either of the two contingencies. The feasibility margin in this case is assessed with a load increase at all 132KV bus bars, while generation is increased at the fossil fuelled stations of Fawley and Didcot, and at the slack 400KV bus bar at Dungeness representing infeed at this point [92].

Single Load Bus Variation Case

This is a realistic power system. We take the PQ bus 4 in the data set of NG 61 bus system to vary, and the both P and Q at bus 4 is varied at equivalently rate. In order to compare easily, the P-V curve of load bus 4 is plotted using CCPF, ECPF and GCPF methods respectively, and the numerical results is also summarized in tables.

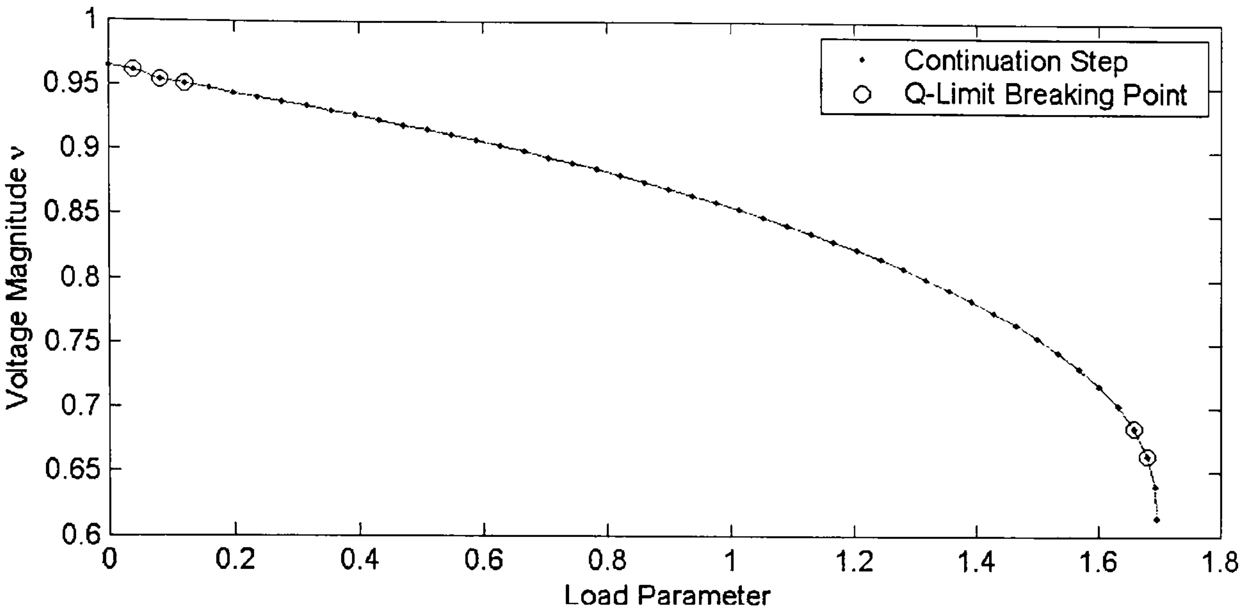


Figure 6-9-a: Varying single load bus of NGC 61 bus system using CCPF

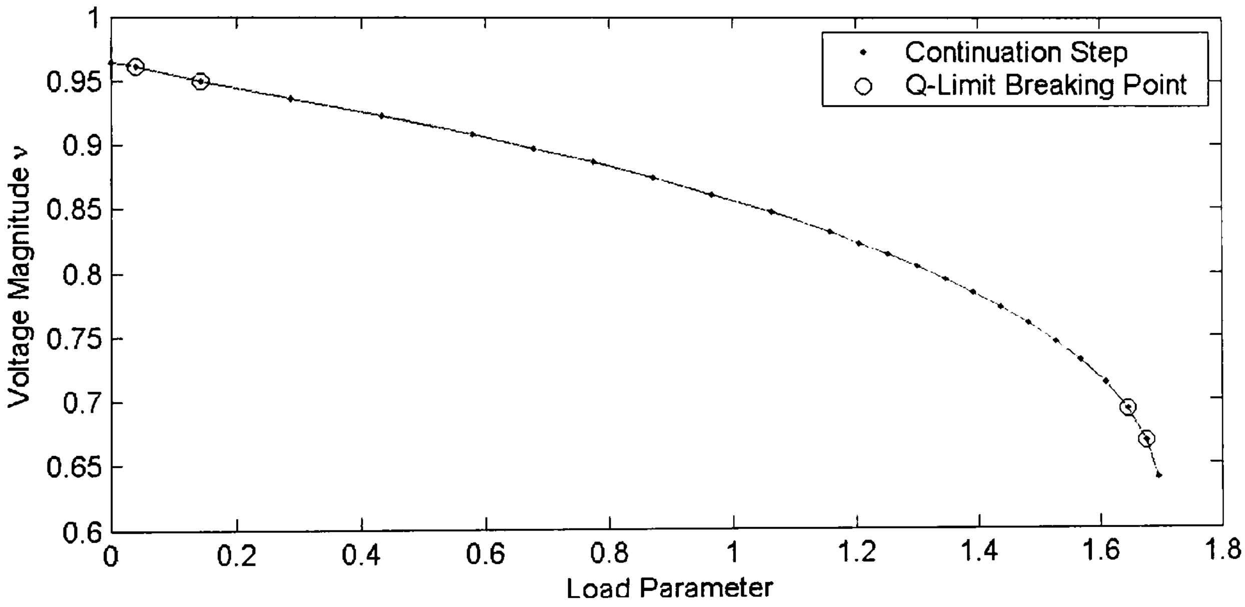


Figure 6-9-b: Varying single load bus of NGC 61 bus system using ECPF

Figure 6-9-a, 6-9-b and 6-9-c illustrate the P-V curve of voltage magnitude of load bus 4 versus load parameter λ. Figure 6-9-a was obtained using the CCPF method, Figure 6-9-b was obtained using ECPF method and Figure 6-9-c was obtained using the novel GCPF method. In Figures 6-9-a, 6-9-b and 6-9-c, we use dots to denote continuation steps and the symbol "o" is used to denote the step where the Q-limit breaking point occurs.

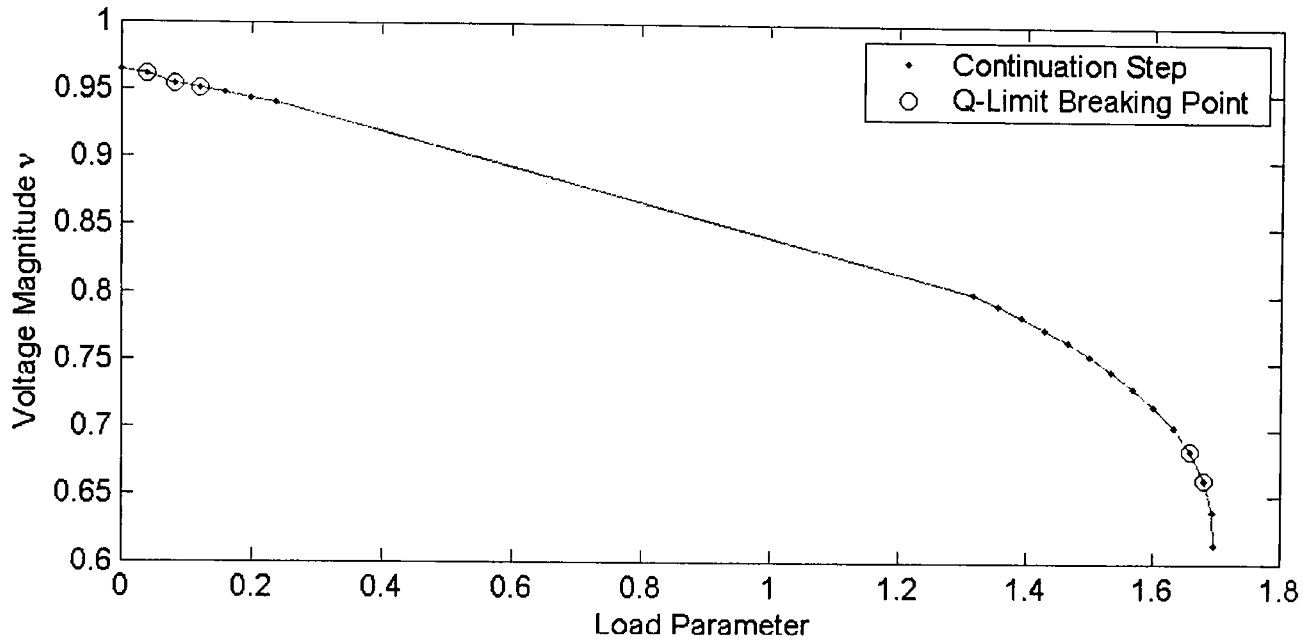


Figure 6-9-c: Varying single load bus of NGC 61 bus system using GCPF

	Bus Type Changes	Continuation Steps	Computation time(s)	Load Parameter
CCPF	8	48	0.5960	1.6952
ECPF	8	25	0.3750	1.6948
GCPF	8	21	0.3520	1.6960

Table 6-9: Results summary of varying single load bus of NGC 61 Bus System

Discussion

This NG 61 bus system is a realistic power system. From the overview of figures and table, we observe that the novel GCPF works much better than CCPF, and is comparable faster than ECPF. From the numerical results in Table 6-9, GCPF method saves 27 and 4 continuation steps comparing to CCPF and ECPF respectively, and reduces 0.6640 and

0.0230 seconds respectively. The NG 61 bus system does not provide the optimal condition to implement GCPF, because the distance between the third and fourth Q-limit breaking point is so long, in other word, the distribution of Q-limit breaking point is not equable, which result in the Lagrange polynomial interpolation formula is not able to predict the Q-limit breaking index accurately. That's why GCPF takes many continuation steps before the fourth Q-limit breaking point.

The P-V curve of Figures 6-9-a and 6-9-c illustrate that there are 5 Q-limit breaking points on the curve, but there are 4 Q-limit breaking points on Figure 6-9-b, the reason is that the ECPF method takes large step size at the beginning of computation, if any two Q-limit breaking points are too close, ECPF changed their bus type from PV to PQ in the same continuation step. The CCPF and GCPF adapt small step size during the whole CPF computation process, so they can secure the calculation accuracy in every single step. The ECPF method can only secure the required calculation accuracy in steps near the bifurcation point. So the GCPF is not only faster, also can keep the same accuracy at every computational step.

Multiple Load Bus Variation Case

In this realistic NG 61 bus system, I pick up the first ten load buses in the data set of NG 61 bus system to vary. They are buses 2 to 11, at which both P and Q is varied at equivalently rate. In order to compare easily, the P-V curve of load bus 4 is plotted using CCPF, ECPF and GCPF methods respectively, and the numerical results are summarized in tables.

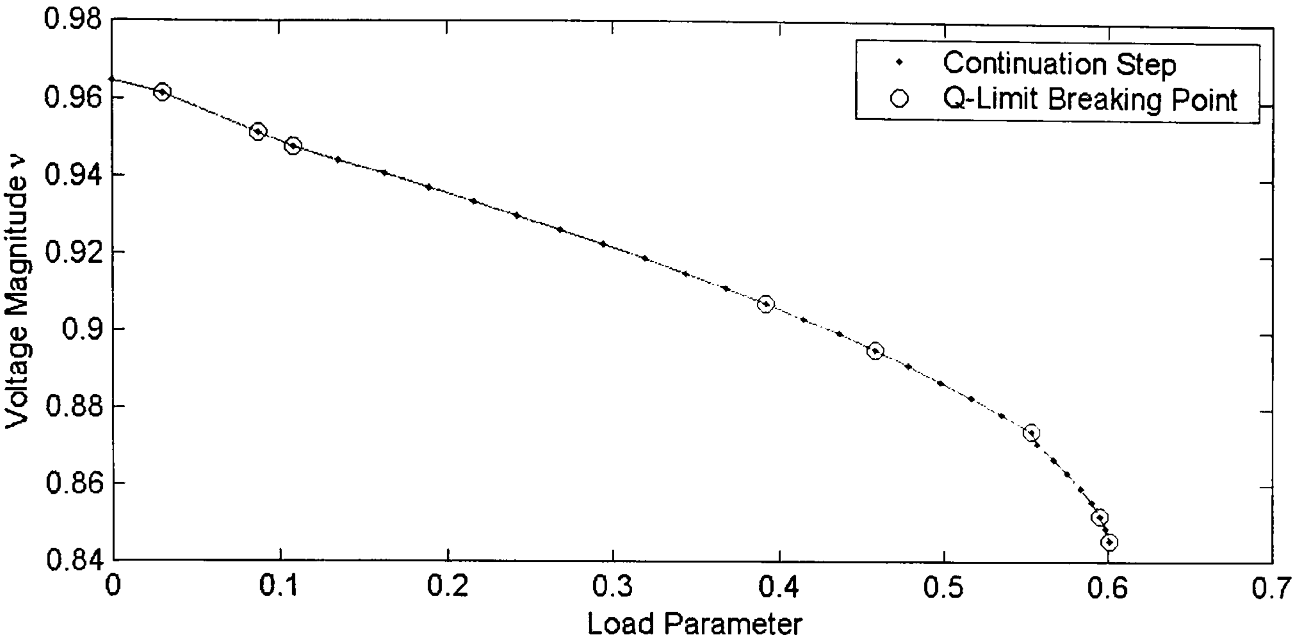


Figure 6-10-a: Varying multiple load buses of NGC 61 bus system using CCPF

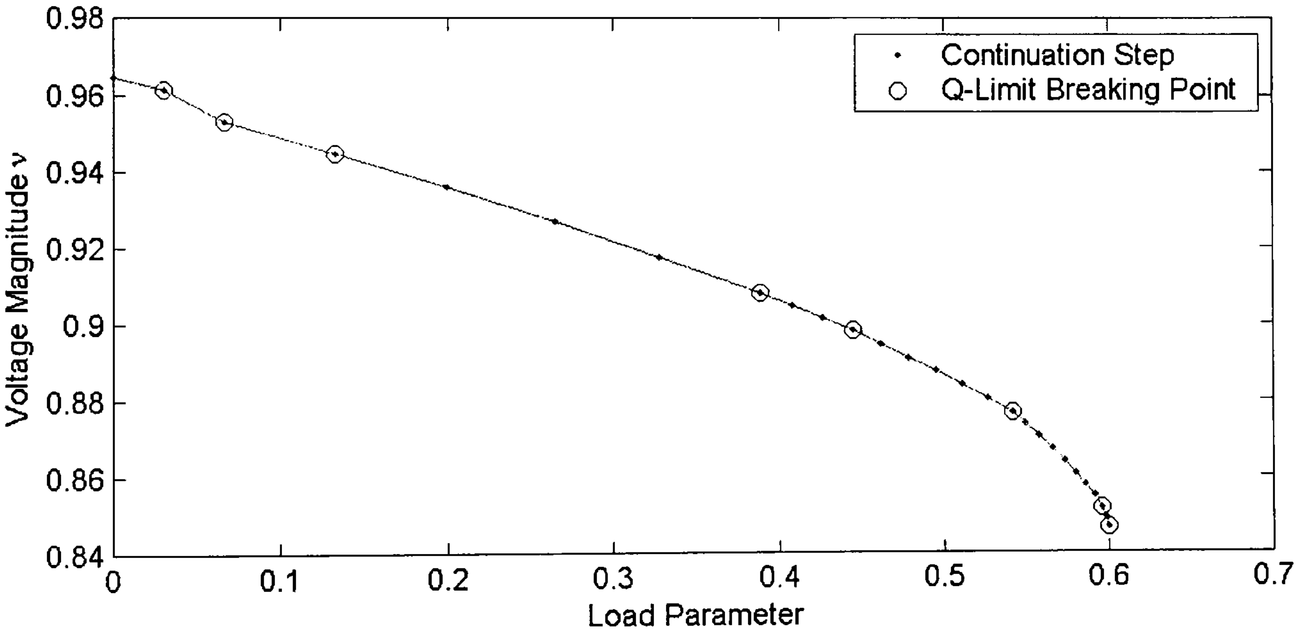


Figure 6-10-b: Varying multiple load buses of NGC 61 bus system using ECPF

Figures 6-10-a, 6-10-b and 6-10-c illustrate the P-V curve of voltage magnitude of load bus 4 versus load parameter λ. Figure 6-10-a was obtained using the CCPF method, Figure 6-10-b was obtained using ECPF method and Figure 6-10-c was obtained using the novel GCPF method. In Figures 6-10-a, 6-10-b and 6-10-c, we use dots to denote continuation steps and the symbol "o" is used to denote the step where the Q-limit breaking point occurs.

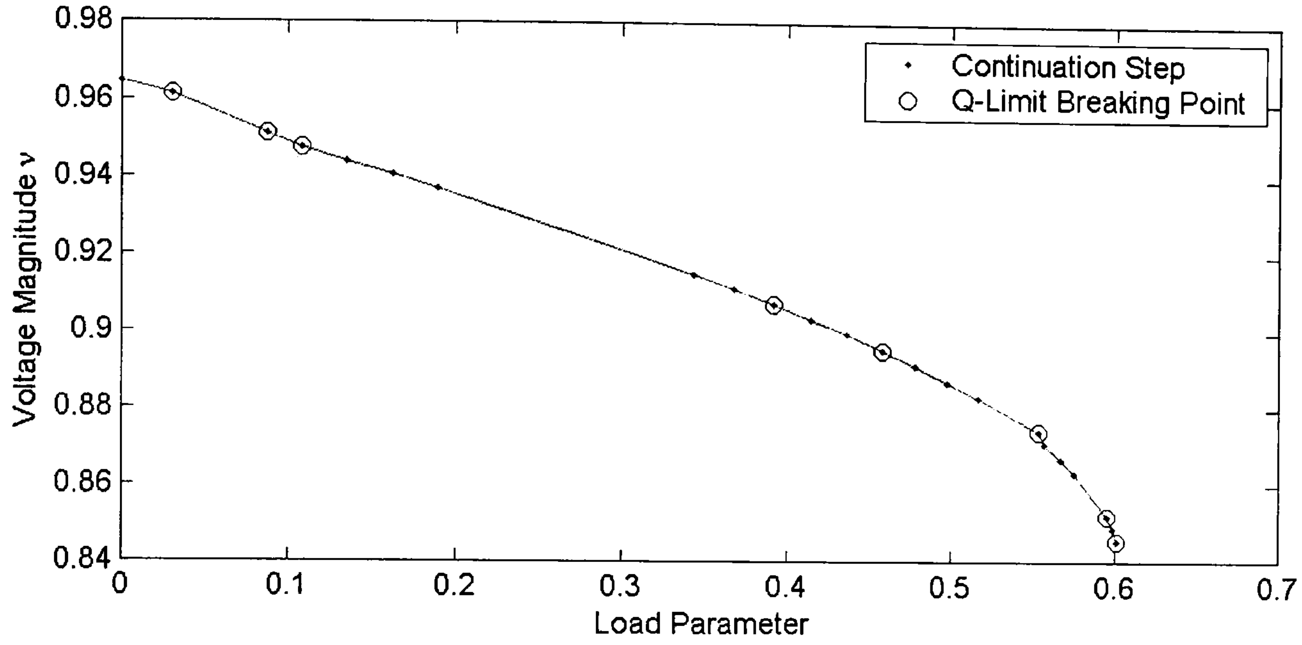


Figure 6-10-c: Varying multiple load buses of NGC 61 bus system using GCPF

	Bus Type Changes	Continuation Steps	Computation time(s)	Load Parameter
CCPF	13	31	0.5278	0.6008
ECPF	13	27	0.4530	0.6004
GCPF	13	23	0.4434	0.6013

Table 6-10: Results summary of varying multiple load buses of NG 61 Bus System

Discussion

From Figure 6-10-a and table 6-10, we see that CCPF method takes 31 continuation steps to reach the bifurcation point, but there are 8 Q-limit breaking points that occur in the whole process, which causes the ratio of Q-limit breaking point to total continuation step is too high, this is another poor condition to implement GCPF method, because GCPF method is not able to pass over any Q-limit breaking point, which makes GCPF be constrained to reduce the number of required steps but this will not affect the ECPF method. So, under this condition the proposed GCPF just presents a general performance, which saves 8 and 4 continuation steps and 0.0844 and 0.0096 seconds comparing to CCPF and ECPF respectively.

We could draw another conclusion that the high ratio of Q-limit breaking point to total continuation steps is not a good condition to implement the proposed GCPF method. In this case, we just get comparable results between CCPF, ECPF and GCPF methods.

6.3 Investigation and Analysis of NG 953 Bus System

This NG 953 bus system is another realistic power system in my research work, which has 96 generators and 397 load buses. The system network has been divided into 5 areas. There are some issues when we test this large realistic system, as a result of the practical characteristics, it is a quite large and complicated power system. Due to the large number of buses in the system, we only consider the multiple load bus variation patterns in this section and we analyse every individual area using CCPF, ECPF and GCPF methods respectively.

Because this practical case has large number of buses and we do not know what mechanism used to number all the buses from the original data. So we couldn't pick up the load buses by regions or location of buses due to the invisible network. For simplicity we choose the varying load buses by the number of PQ bus in the data set of each area, the first 25 PQ buses are chosen to increase as the multiple load bus variation pattern. In future work, I will draw out a visible network diagram for NG 953 system, and implement the multiple patterns by regions to analyse the network characteristics further.

Multiple Load Bus Variation Case in Area One

Area one has 11 generators, 1 slack bus and 93 PQ buses, the first 25 PQ buses are chosen to vary, they are bus number 1 to 23, 28 and 29, at which both P and Q are varied at equivalently rate. In order to compare easily, the P-V curve of load bus 13 is plotted using CCPF, ECPF and GCPF methods respectively, and the numerical results are summarized in tables.

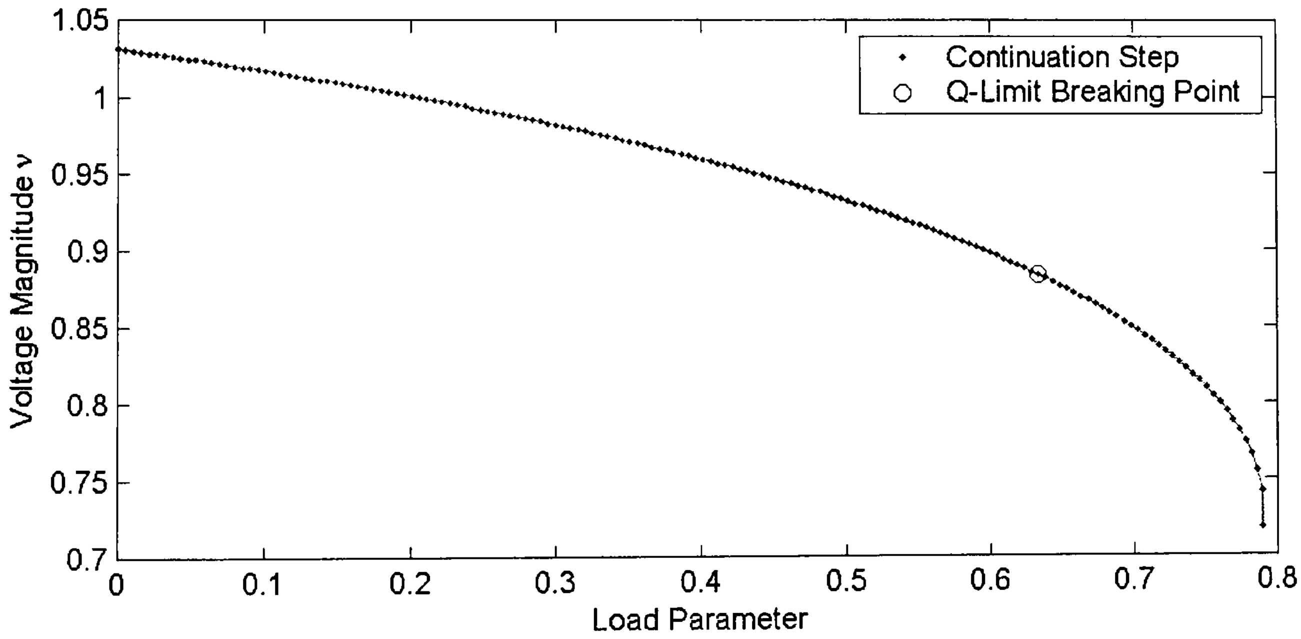


Figure 6-11-a: Varying multiple load buses in area one of NG 953 using CCPF

Figures 6-11-a, 6-11-b and 6-11-c illustrate the curve of voltage magnitude of load bus 13 versus load parameter λ. Figure 6-11-a was obtained using the CCPF method, Figure

6-11-b was obtained using ECPF method and Figure 6-11-c was obtained using the novel GCPF method. In Figures 6-11-a, 6-11-b and 6-11-c, we use dots to denote continuation steps and the symbol “o” is used to denote the step where the Q-limit breaking point occurs.

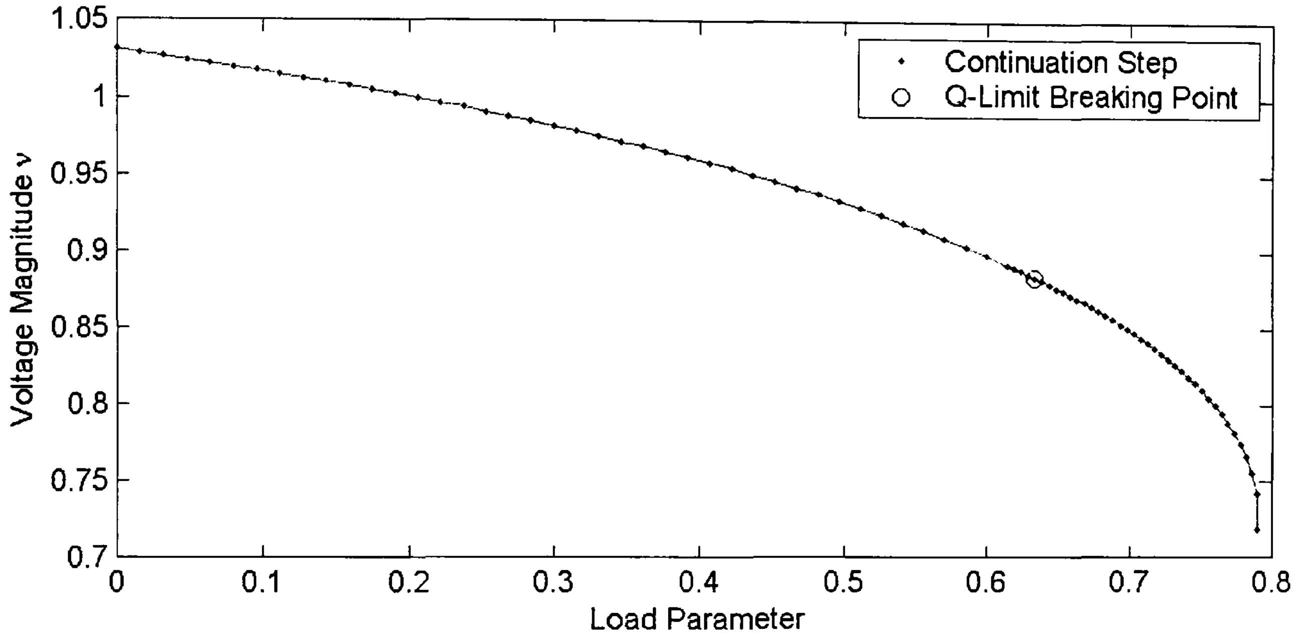


Figure 6-11-b: Varying multiple load buses in area one of NG 953 using ECPF

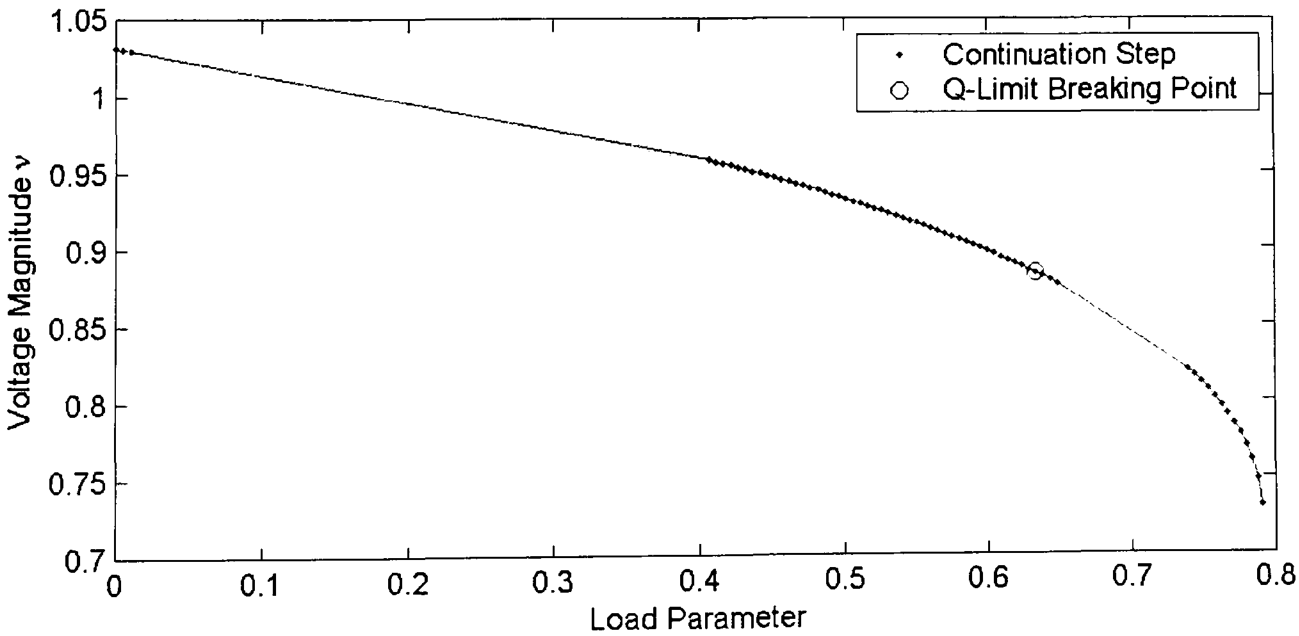


Figure 6-11-c: Varying multiple load buses in area one of NG 953 using GCPF

	Bus Type Changes	Continuation Steps	Computation time (s)	Load Parameter
CCPF	1	159	23.1954	0.7901
ECPF	1	79	15.0064	0.7901
GCPF	1	66	10.5502	0.7906

Table 6-11: Results summary of varying multiple load buses in area one of NG953

Discussion

In this case, there is an only one generator break its Q-limit, which is not a good condition to implement the novel GCPF method actually. From Figure 6-11-c, we can observe GCPF takes too many steps to reach the only Q-limit breaking point, which means the Lagrange polynomial interpolation technique does not predict the Q-limit breaking index accurately, but this approach still guides the continuation steps to the Q-limit breaking point as fast as it can and saves large number of steps required. This generator of Q-limit breaking was the slack bus, after changing it to PQ bus type, the next PV bus in the data set that will be the new slack bus in GCPF algorithm.

Under this condition, our GCPF method saves 93 and 13 computational steps, 12,6452 and 4.4562 seconds in computational time comparing to CCPF and ECPF respectively.

Multiple Load Bus Variation Case in Area Two

Area two has 10 PV buses and 156 PQ buses, the first 25 PQ buses are selected to vary, they are buses 107 to 112 and 114 to 132, at which both P and Q are varied at equivalently rate. The P-V curve of bus 111 is first plotted using CCPF method.

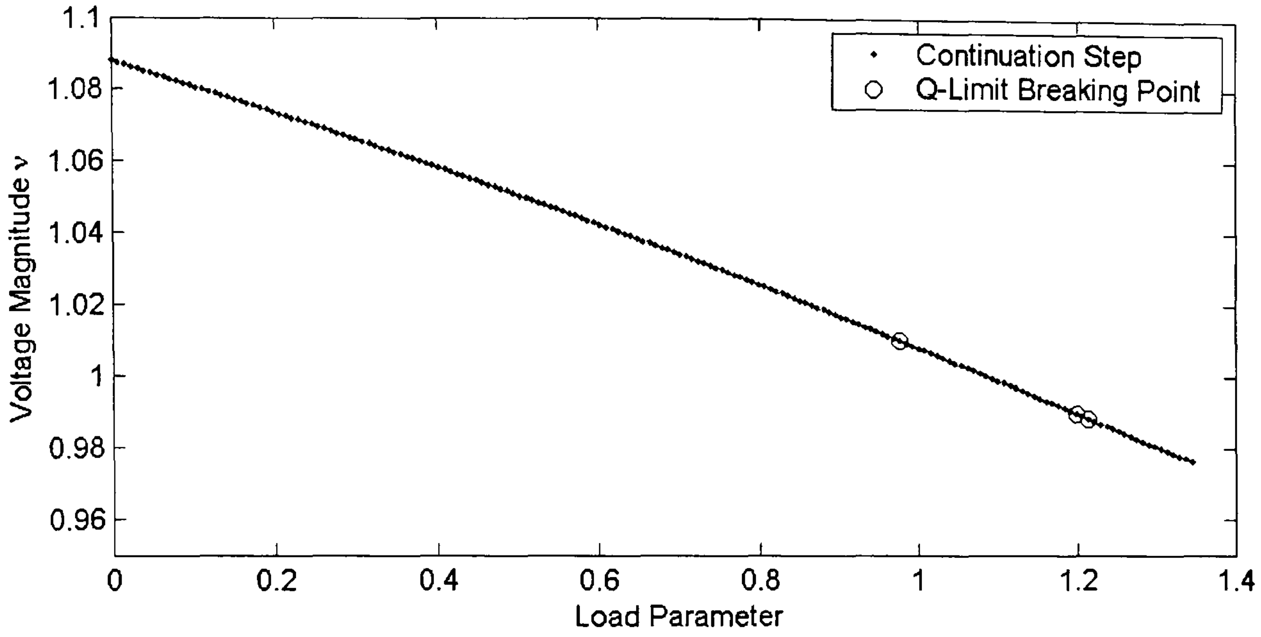


Figure 6-12: Failure in varying multiple load buses in area two of NG953

When we implement the CCPF method in area two, the solution procedure suddenly breaks down. Figure 6-12 illustrates the P-V curve in this situation that the computation process suddenly terminated at the last step.

Voltage collapse can occur in a power system following a progressive decline at the bifurcation point, usually due to load increasing. A lack of adequate reactive power compensation in a power system has been recognized as a major contributing factor in a voltage collapse process. As we know, once a reactive power compensator has reached its limit, it can no longer adjust the voltage. Consequently, large increases in load may result in sudden voltage collapse. Although there are some other factors that influence the voltage collapse process, in this paper we have focussed on reactive power compensators reaching their Q-limit.

Therefore, in this research work we adopted a strategy that removes largest load bus first, then the second largest load bus, and so on in this manner until the algorithm is able to determine a definite saddle-node bifurcation point. We have reported the results of this strategy in table 6-2.

Varying bus list	Generators	steps	Load parameter	time (s)
107to112,114 to132	1,18,81,86,15	183	0.5173	16.1720
Remove bus 109	1,18,86,81	178	0.7738	15.3910
109,118	1,18,86,81	177	0.8994	15.4530
109,118,116	1,86,81,18	177	1.0706	15.5630
109,118,116,120	1,18,86,81	170	1.2082	15.1720
109,118,116,120,123	1,18,81,86	179	1.4751	15.7500
109,118,116,120,123 122	1,18,81,86	188	1.8201	16.5000
109,118,116,120,123 122,128	1,81,86	189	2.2112	16.5470
109,118,116,120,123 122,128,130	1,81,86	178	2.5543	15.7190
109,118,116,120,123 122,128,130,124,125	18,86,1,15	148	3.2779	14.2340

Table 6-12: Results of removal strategy for area two of NG 953

The top row entry in the “Varying Bus List” column specifies the node numbers of load buses that are varied. The second row entry to the bottom specifies the node numbers of buses removed from the list. The bus numbers are given in the sequence that they are removed. The column “Generators” refers to the number of generators that break their Q-limit, where the numbers are given in the Q-limit breaking sequence.

From the results in Table 6-12, the bus 109 has the largest load demand that is 467.50 MW (active power) and 39.00 Mvar (reactive power), the bus 118 is the second largest load bus that is 130.20 MW (P) and 42.30 (Q), and so on there are 10 buses has been removed form the Bus Varying List in order finally. The other 15 buses left in varying list to keep increasing in order to determine a definite saddle-node bifurcation point.

In this investigation, I observe that these large load buses are very sensitive to the system stability and result in sudden voltage collapse. We define the buses causing sudden voltage collapse as "weakness bus", if the buses locate in an individual regain, this regain will be defined as "weakness area". Since now, we are able to implement the comparison work between CCPF, ECPF and GCPF methods for the area two of NG 953 case.

Multiple Load Buses Variation Case in Area Two

In order to compare easily, the P-V curve of load bus 111 is plotted using CCPF, ECPF and GCPF methods respectively, and the numerical results is also summarized in tables.

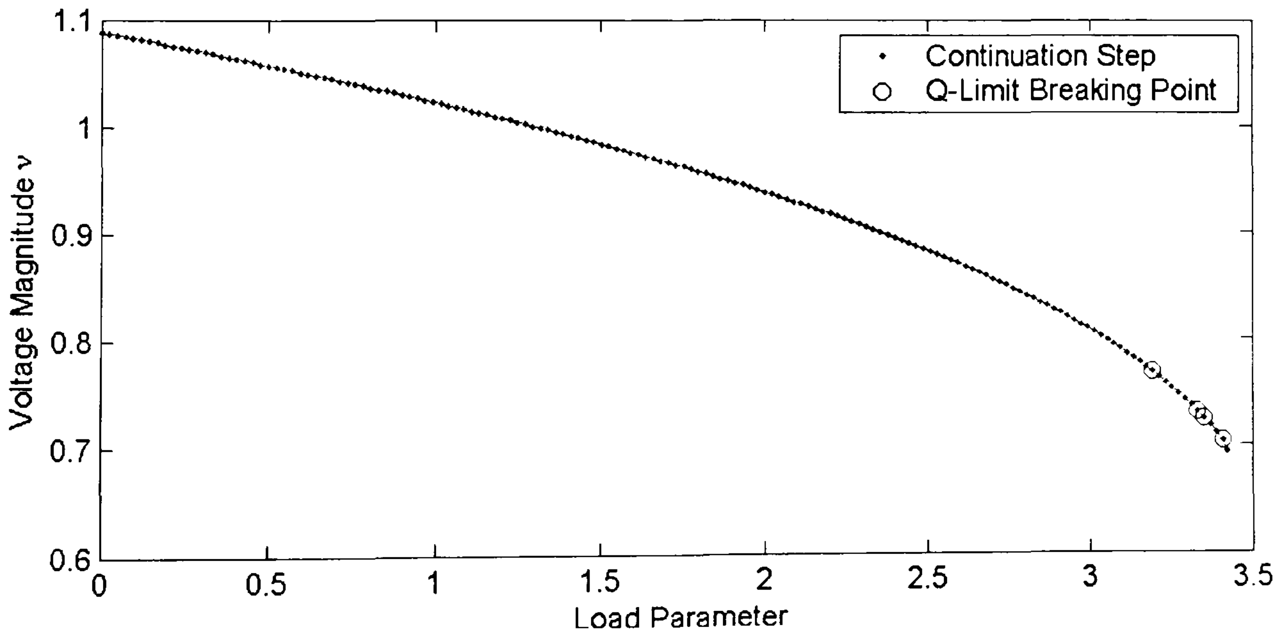


Figure 6-13-a: Varying multiple load buses in area two of NG 953 using CCPF

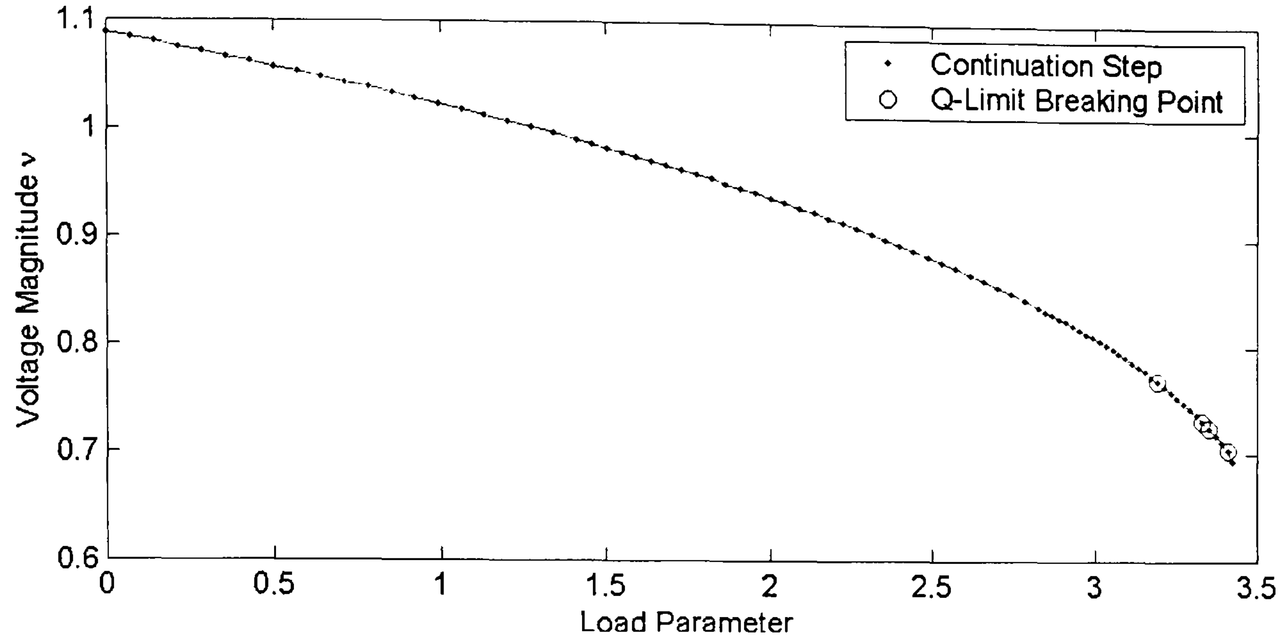


Figure 6-13-b: Varying multiple load buses in area two of NG 953 using ECPF

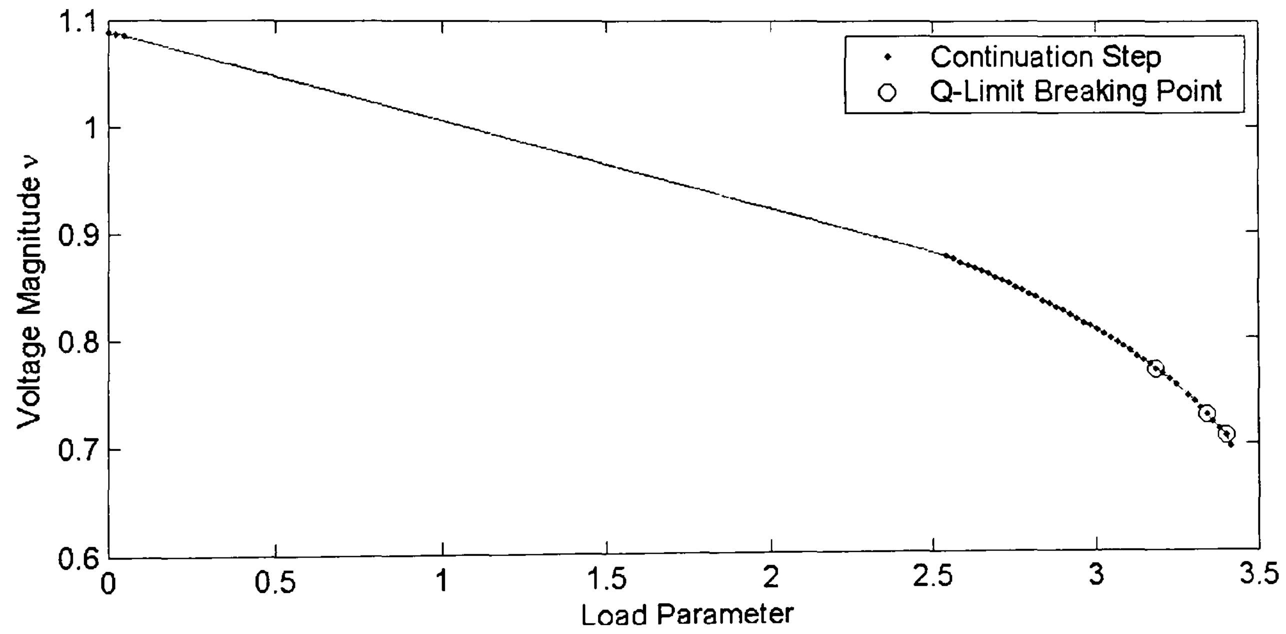


Figure 6-13-c: Varying multiple load buses in area two of NG 953 using GCPF

Figure 6-13-a, 6-13-b and 6-13-c illustrate the curve of voltage magnitude of load bus 111 versus load parameter λ. Figure 6-13-a, 6-13-b and 6-13-c were obtained using CCPF. ECPF and the novel GCPF method respectively. In figures, we use dots to denote continuation steps and the symbol "o" is used to denote the step where the Q-limit breaking point occurs.

	Bus Type Changes	Continuation Steps	Computation time(s)	Load Parameter
CCPF	4	155	22.7126	3.4215
ECPF	4	83	17.4090	3.4215
GCPF	4	48	9.7904	3.4274

Table 6-13: Results summary of varying multiple load buses in area two of NG953

Discussion

From Figures 6-13-a, 6-13-b and 6-13-c, we observe that the Q-limit breaking points are centralized in the end region of the curve. GCPF method takes many computational steps to reach the first Q-limit breaking point, which may caused by the poor prediction of Q-limit breaking index. From the numerical results in Table 6-13, the proposed GCPF works more efficient than others, GCPF takes 107 and 35 computational steps less than CCPF and ECPF respectively, and saves 12.9222 and 7.6186 seconds in computational time respectively. The results indicate the novel GCPF does improve the CPF method, and is faster than the improved ECPF method.

Multiple Load Buses Variation Case in Area Three

Area three has 47 PV buses and 311 PQ buses, I choose the first 25 buses to vary, and the bus numbers are 263 to 287, at which both P and Q are varied. In order to compare easily, the P-V curve of load bus 264 is plotted using CCPF, ECPF and GCPF methods respectively, and the numerical results is also summarized in tables.

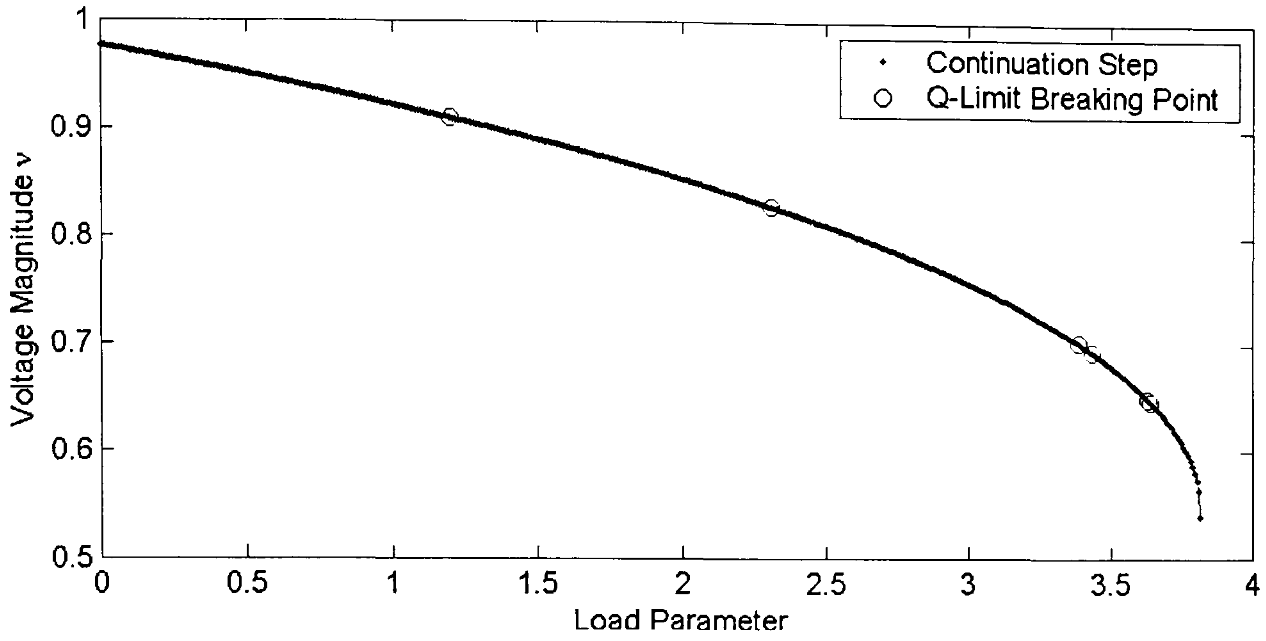


Figure 6-14-a: Varying multiple load buses in area three of NG 953 using CCPF

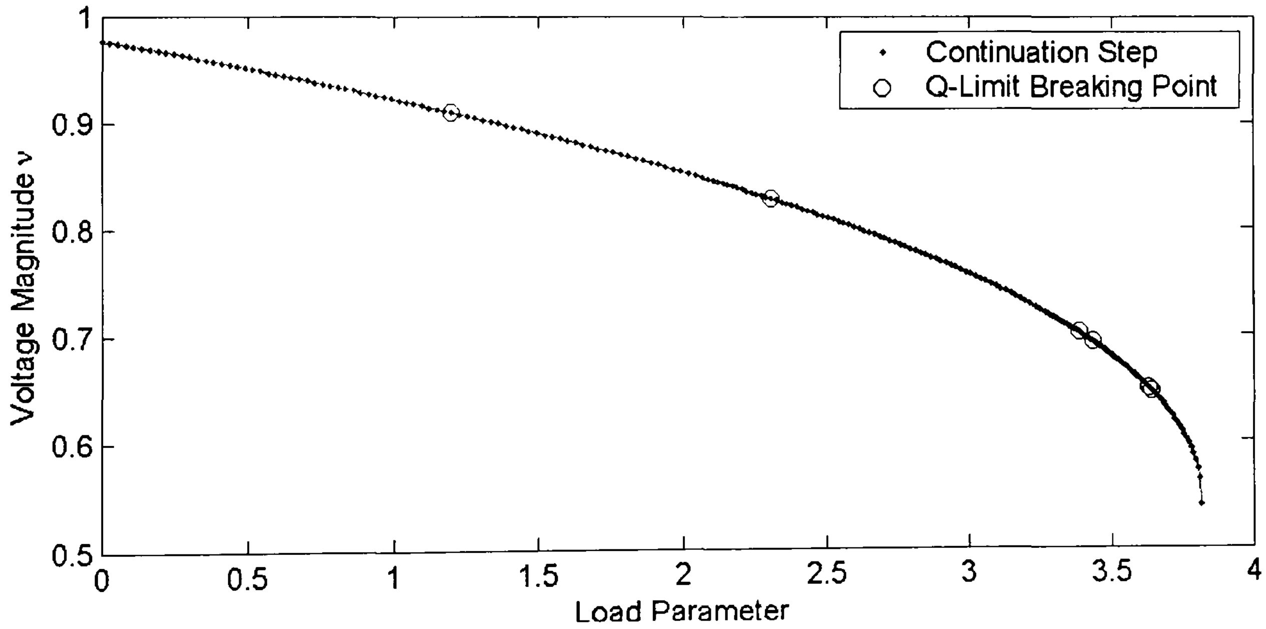


Figure 6-14-b: Varying multiple load buses in area three of NG 953 using ECPF

Figure 6-14-a, 6-14-b and 6-14-c illustrate the curve of voltage magnitude of load bus 264 versus load parameter λ. Figure 6-14-a, 6-14-b and 6-14-c were obtained using CCPF. ECPF and the novel GCPF method respectively. In figures, we use dots to denote continuation steps and the symbol “o” is used to denote the step where the Q-limit breaking point occurs.

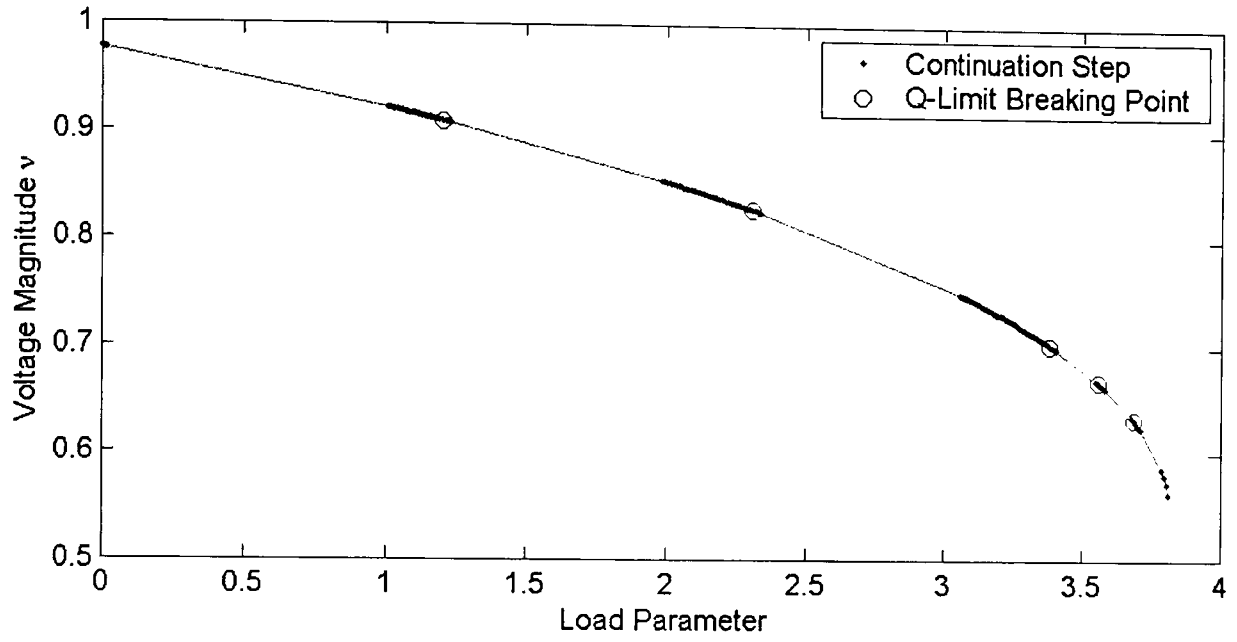


Figure 6-14-c: Varying multiple load buses in area three of NG 953 using GCPF

	Bus Type Changes	Continuation Steps	Computation time(s)	Load Parameter
CCPF	8	438	57.1436	3.8184
ECPF	8	214	29.2752	3.8184
GCPF	8	128	18.8066	3.8124

Table 6-14: Results summary of varying multiple load buses in area three of NG953

Discussion

From Figure 6-14-c, the P-V curve was obtained by applying the GCPF method to area three of NG 953 bus system. It is obvious from the figure that the curve can be divided into several sections this is because the continuation steps have been effectively guided by the predicted Q-limit breaking indices, in this way the GCPF method is able to significantly reduce large number of computational steps required.

From the numerical results in Table 6-14, this case provides a very good condition to implement the GCPF method, the Q-limit breaking points locate on the curve evenly, and the Q-limit breaking indices guide the continuation steps efficiently to reach the breaking point in a fast way. The result of this case is also the optimal result we expected. From the numerical results in Table 6-14, we can find that the novel GCPF method takes 310 and 86 computational steps less than the CCPF and ECPF methods respectively, and saves 38.3370 and 10.4686 seconds in computational time comparing with the CCPF and ECPF methods respectively.

Multiple Load Buses Variation Case in Area Four

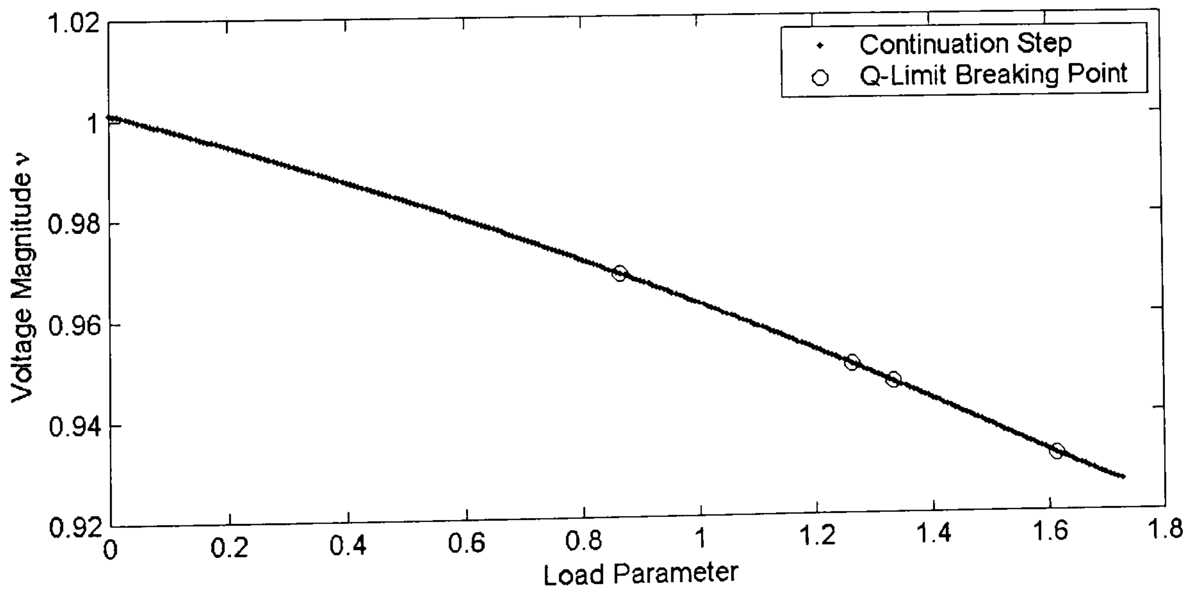


Figure 6-15: Failure in varying multiple load buses in area four of NG953

Area four has 21 PV buses and 273 PQ buses totally. The first 25 PQ buses are selected to vary, they are buses 621 to 635, 643 to 646 and 648 to 653, at which both P and Q are varied. The P-V curve of bus 650 is plotted using CCPF method.

When we implement the CCPF method in area four of NG 953, the computation has the similar situation to that happened in area two that is the solution procedure suddenly breaks down. Figure 6-15 illustrates the P-V curve in this case.

A Lack of adequate reactive power resources in a power system has also been recognized as a major contributing factor causing system sudden voltage collapse in this case. The large load increasing resulted in this situation. With regard to this issue, the removal strategy has also been applied in order to determine a definite saddle-node bifurcation point in this case. The largest load bus has been removed first, and then removed the second largest load bus, and so on in this manner until a reliable bifurcation point passed. The results of this removal strategy are presented in Table 6-15.

Varying bus list	Generators	steps	Load parameter	time (s)
621to635,643to646, 648 to 653	1,86,81,2	258	1.7276	22.5160
Remove bus 627	1,81,2,86	293	2.8113	25.5930
627,648	1,2,81,86	333	4.3811	29.0310
627,648,629	1,2,81,86	320	5.3433	28.1526
627,648,629,628	1,81	260	5.7363	23.3440

Table 6-15: Results of removal strategy for area four of NG 953

The top row entry in the “Varying Bus List” column of Table 6-15 specifies the node numbers of load buses that are varied. The following rows entry specify the node numbers of buses removed from the list. The bus numbers are given in the sequence that they are removed. The column “Generators” refers to the number of generators that break their Q-limit, where the numbers are given in the Q-limit breaking sequence.

From the results in Table 6-15, the four largest buses have been removed form the list in order totally. The largest load bus 627 is removed first, at which the active power is 188.60 MW and the reactive power is 104.60 Mvar, and so on until the varying bus list retains the other 21 buses to increase in order to determine the saddle-node bifurcation point. We can obtain a reliable voltage collapse index after taking out four largest load buses from the varying list. Figure 6-16-a illustrates the P-V curve obtained from the last successful attempt to determine the saddle-node bifurcation point. These four large load buses are quite sensitive to the voltage stability of NG 953 bus power system.

We can now implement the comparison work between the CCPF, ECPF and GCPF methods for the case of area four. In order to compare easily, the P-V curve of load bus 650 is plotted using CCPF, ECPF and GCPF methods respectively, and the numerical results are summarized in tables.

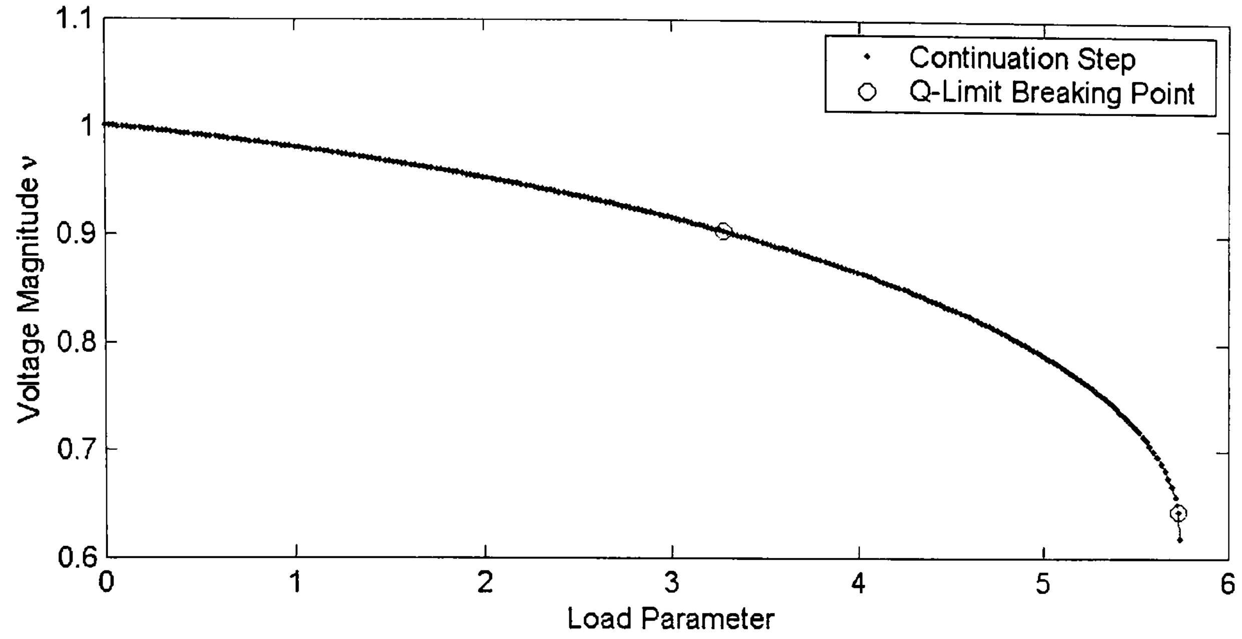


Figure 6-16-a: Varying multiple load buses in area four of NG 953 using CCPF

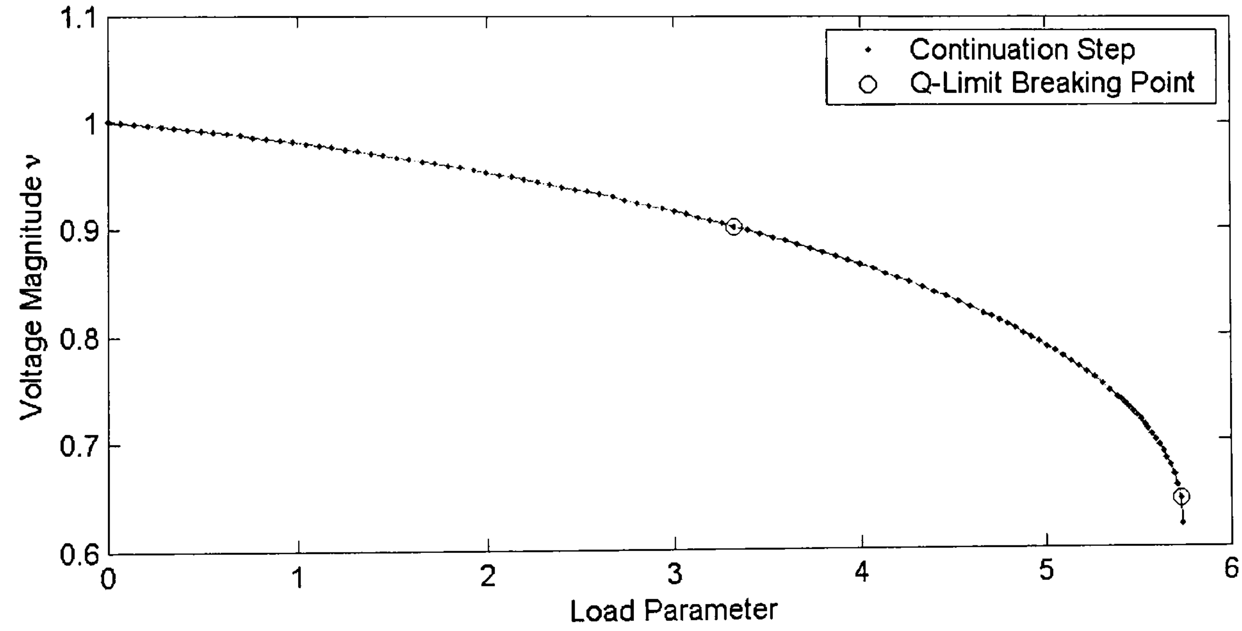


Figure 6-16-b: Varying multiple load buses in area four of NG 953 using ECPF

Figure 6-16-a, 6-16-b and 6-16-c illustrate the curve of voltage magnitude of load bus 650 versus load parameter λ. Figure 6-16-a, 6-16-b and 6-16-c were obtained using CCPF. ECPF and the novel GCPF method respectively. In figures, we use dots to denote continuation steps and the symbol "o" is used to denote the step where the Q-limit breaking point occurs.

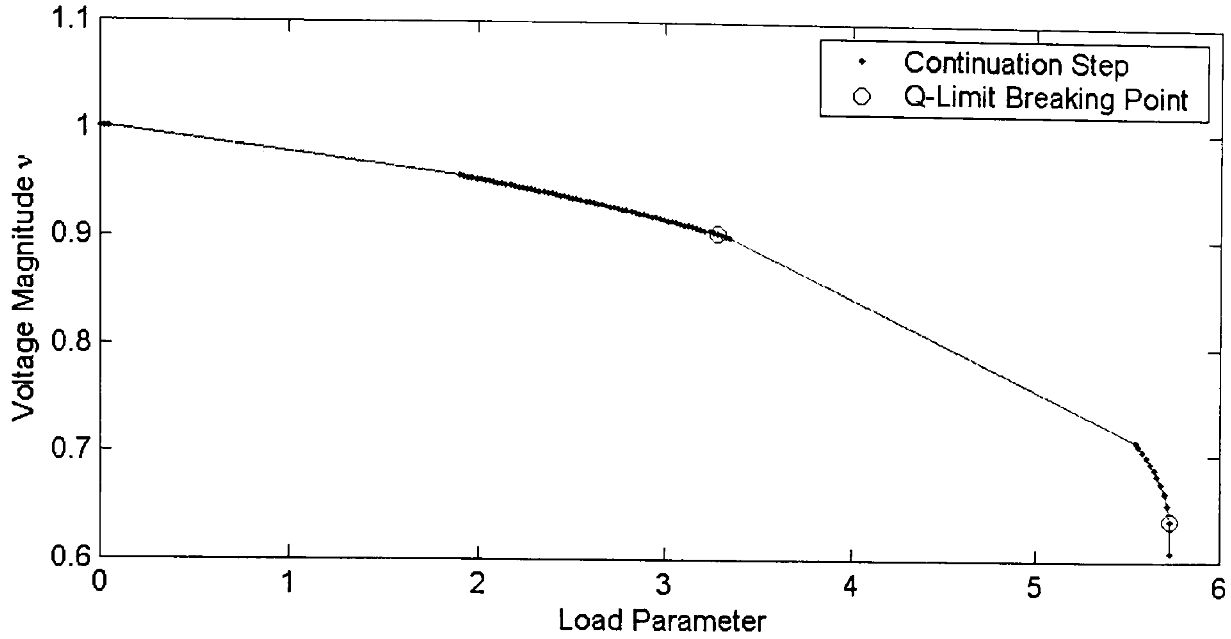


Figure 6-16-c: Varying multiple load buses in area four of NG 953 using GCPF

	Bus Type Changes	Continuation Steps	Computation time(s)	Load Parameter
CCPF	2	260	24.5408	5.7363
ECPF	2	105	15.3406	5.7365
GCPF	2	81	9.1096	5.7337

Table 6-16: Results summary of varying multiple load buses in area four of NG953

Discussion

This case has a similar situation to area two, we adopt removal strategy to take some weakness buses out of the varying bus list, and then the computation is able to pass through the definite saddle-node bifurcation point. There are two generators that break their Q-limit, the distribution of these two Q-limit breaking points is good to implement novel GCPF method, but it still takes many steps to reach the first Q-limit breaking point. From the numerical results in table 6-16, it is easy to see that GCPF is much faster than another two methods, the GCPF takes 179 and 24 continuation steps less than the CCPF and ECPF respectively and saves 15.4312 and 6.2310 seconds in computational time compared with the CCPF and ECPF respectively.

Multiple Load Buses Variation Case in Area Five

Area five has 6 PV buses and 33 PQ buses, and is the smallest individual area compared with the other areas. But the load demand and generation is relative larger, which means that the power transfer in this area is heavy and centralized. So, area five is more sensitive to the system voltage stability with the load varying, and is easier to cause voltage collapse of system. As a result, buses 919, 920, 927 to 929, 936, 939, 942, 943, 946 to 948, 950 and 951 of which the active power demand is not bigger than 50 MW are selected to vary as the multiple load buses variation pattern specially. The total active power and reactive power of these buses is 234.1 MW and 68 Mvar.

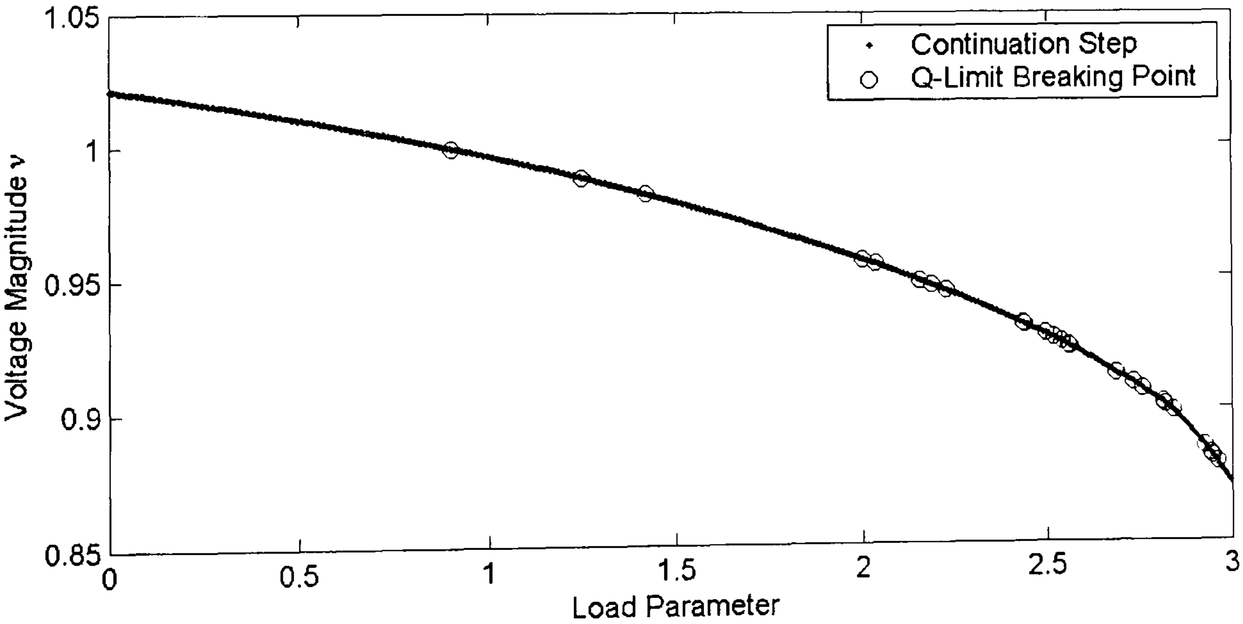


Figure 6-17-a: Varying multiple load buses in area five of NG 953 using CCPF

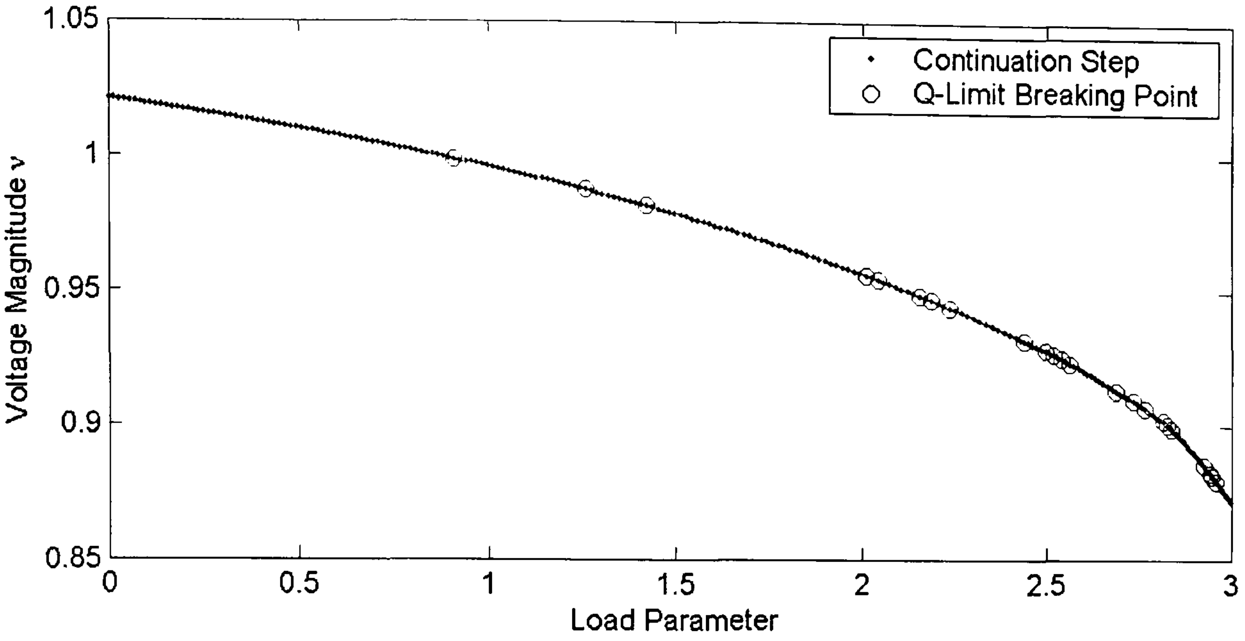


Figure 6-17-b: Varying multiple load buses in area five of NG 953 using ECPF

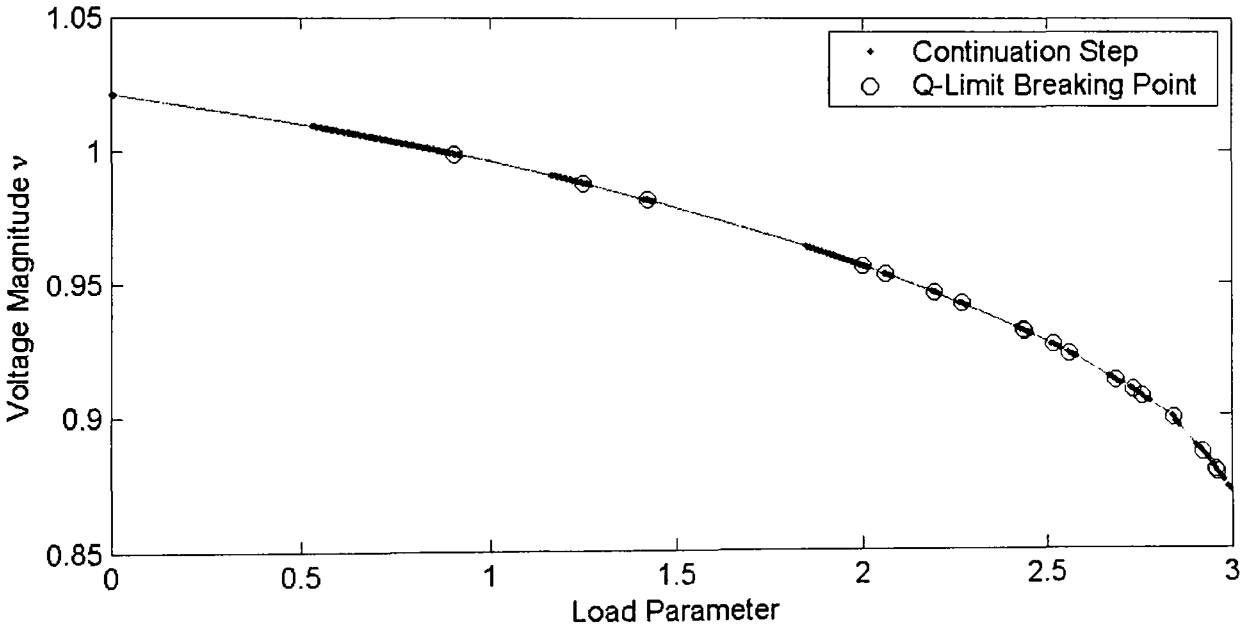


Figure 6-17-c: Varying multiple load buses in area five of NG 953 using GCPF

In order to compare easily, the P-V curve of load bus 928 is plotted using CCPF, ECPF and GCPF methods respectively, and the numerical results are summarized in Table 6-17. Figures 6-17-a, 6-17-b and 6-17-c illustrate the curve of voltage magnitude of load bus 928 versus load parameter λ . Figure 6-17-a, 6-17-b and 6-17-c were obtained using CCPF, ECPF and the novel GCPF method respectively. In figures, we use dots to denote continuation steps and the symbol "o" is used to denote the step where the Q-limit breaking point occurs.

	Bus Type Changes	Continuation Steps	Computation time(s)	Load Parameter
CCPF	27	589	53.2780	2.9979
ECPF	27	229	24.5526	2.9973
GCPF	27	216	20.8594	2.9965

Table 6-17: Results summary of varying multiple load buses in area five of NG953

Discussion

In this case, there are many generators that reached their Q-limits, which indicates area five is heavy loaded and quite sensitive to the whole network. This situation is quit similar to the NG 61 bus system, where too many Q-limit breaking points during computation are not a good condition to implement our GCPF method, which will reduce the efficiency of GCPF. From Figure 6-17-c, Lagrange polynomial Q-limit predictor did guide the continuation steps to every Q-limit breaking point one by one efficiently. However, as our GCPF algorithm is based on Q-limit prediction, it is not able to pass over any Q-limit breaking points, and 3 computational steps must be computed for data collection of each Q-limit index, which are limiting factors on the efficiency of GCPF method. From the numerical results in Table 6-17, the GCPF method indicates the comparable performance to the improved ECPF method. We just obtained a little profit compared with the ECPF method. However, the novel GCPF is much more efficient than the CCPF method, and saves large number of steps and computing time required.

From the numerical analysis, the novel GCPF takes 373 continuation steps and 32.4186 seconds computing time less than CCPF method, and it takes 13 steps and 3.6932 seconds less compared with the improved ECPF method.

6.4 Summary

Test Systems	Bus Type Changes	Continuation Steps			Computational Time (s)			Speed Up (%)	
		CCPF	ECPF	GCPF	CCPF	ECPF	GCPF	CCPF	ECPF
IEEE9(S)	1	67	39	24	0.5684	0.3780	0.2406	58	36
IEEE9(M)	1	48	29	20	0.2420	0.1720	0.1325	45	23
IEEE30(S)	5	166	97	35	0.8280	0.5470	0.4060	51	26
IEEE30(M)	4	78	47	35	0.6720	0.4840	0.3750	44	23
IEEE118(S)	17	243	141	71	3.2184	2.2575	1.5620	51	31
IEEE118(M)	20	206	132	92	2.8280	2.2196	1.6720	41	25
NG61(S)	8	48	25	21	0.5960	0.3750	0.3520	41	6
NG61(M)	13	31	27	23	0.5278	0.4530	0.4434	16	2
NG953(M01)	1	159	79	66	23.1954	15.0064	10.5502	55	30
NG953(M02)	4	155	83	48	22.7126	17.4090	9.7904	57	44
NG953(M03)	8	438	214	128	57.1436	29.2752	18.8066	67	36
NG953(M04)	2	260	105	81	24.5408	15.3406	9.1096	63	41
NG953(M05)	27	589	229	216	53.2780	24.5526	20.8594	61	15

Table 6-18: Result comparison of the CCPF, ECPF with GCPF in all cases

In Table 6-18, the “M” or “S” that follows the case name denotes the load pattern as follows: multiple load bus case and single load bus case respectively, the number following “M” in the NG953 case denotes the number of the area. “bus type changes” refers to how many generator buses reached their Q-limit and have been changed from PV bus type to PQ bus type and the "Speed Up" refers to the percentage speed up of the computation time of the GCPF method compared against the CCPF and ECPF methods.

Having illustrated how the novel GCPF method works for each case, we have now demonstrated the effectiveness of the new algorithm. We summarize a comparison of results for all test cases in Table 6-18. It is relatively simple to compare and analyze the number of continuation steps and computational times as presented in Table 6-18. In all but one case, the GCPF shows the best performance and is much faster than CCPF or ECPF. However, in the NGC 61(M) test system GCPF is comparable with ECPF method. This may be attributed to the large number of Q-limit breaking points that occur in different steps during the continuation procedure, but it is still faster than the CCPF.

From Table 6-18, we observe that

1. The proposed GCPF is generally computationally faster than the standard CPF and ECPF;
2. All the methods can compute the bifurcation point for each case under the required accuracy;
3. Q-limit breaking before the bifurcation point is quite common among the cases we tested;
4. The performance of GCPF is depending on the quantity and distribution of Q-limit breaking points.
5. CCPF is a reliable and stable method to compute the critical point.

Chapter 7: Comparison and Analysis of Linear Solvers in Power Flow

In this section, I have made plenty of experiments to compare the performance between direct method and iterative methods. In order to obtain meaningful comparisons, it is necessary to ensure the same general testing conditions for all of the methods. The column approximate minimum degree (AMD) algorithm is used in incomplete LU (ILU) matrix factorization pre-conditioner for the direct and iterative methods. The AMD ordering scheme benefits the iterative methods as well as the direct method. The ILU factorization method used in iterative methods as a pre-conditioner reduces the number of iterations and speeds up the convergence procedures.

7.1 Introduction and Motivation

The power flow is a very well known algorithmic problem which is usually solved by the Newton-Raphson solution method. The linearization of the nonlinear power flow equations produces a linear system to be solved by an appropriate linear solver. All the power flow algorithms based upon the Newton-Raphson method have in common that one large or two smaller sizes linear systems of equations must be solved during each Newton-Raphson iterative step. The characteristics of these linear systems of equations are sparse, asymmetric and indefinite, due to these characteristics some mathematicians have derived several methods to solve this type of linear system of equations, and in our research work we may consider the generalized minimum residual method (GMRES), bi-conjugate gradient method (BiCG) and bi-conjugate gradient stabilized method (BiCGStab). Practical usage of these iterative methods is only reached when preconditioning is applied to the linear system of equations. Good preconditioning will group the eigenvalues of the transformed linear system matrix together and will thus result in faster convergence, which is fully discussed in this chapter.

Solving a linear equation on the form Ax = b plays an important role in power flow calculation, and uses the most computational time in a power flow. CPF is a process that computes many power flow solutions, so an efficient linear solver is necessary to speed up a CPF method. In this section, studies compare the performance of direct and iterative solvers. It is well established that iterative methods for solving large sparse sets of linear equations can naturally profit from parallelization and factorization.

These meaningful comparisons are conducted on the following network cases: IEEE 30, IEEE 118, IEEE 300, Poland (POL) 2736, National Grid (NG) 61 and NG 953 bus systems.

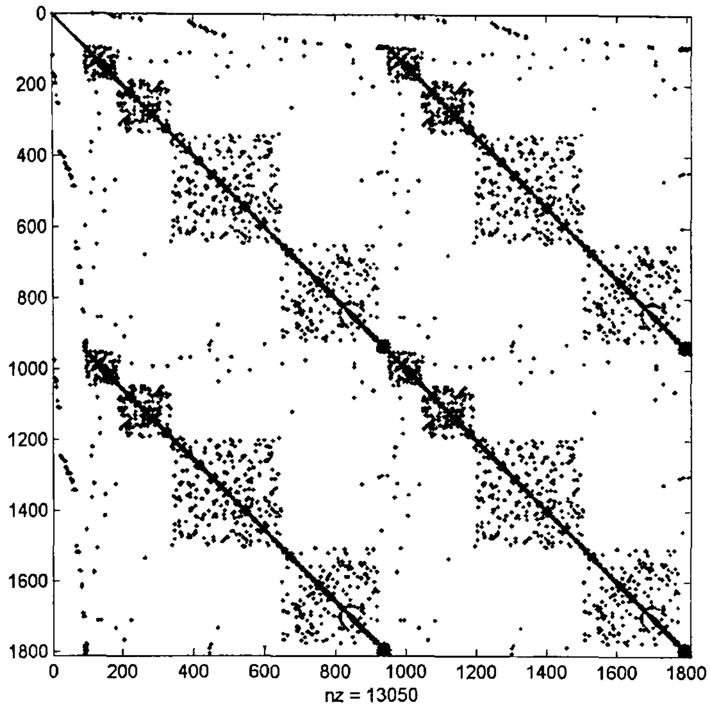
7.2 Column Approximate Minimum Degree (AMD) Ordering Scheme

AMD ordering scheme can preserve the sparsity of the matrix, which affect the eigenvalue spectrum effectively and reduce the convergence rate of the iterative methods. Although sparsity is preserved, ordering may affect the size of the norm of R ($A = M + R$,where M is the pre-conditioner, A is the coefficient matrix and R is the remainder), the number of iterations in iterative method is almost related to the norm of the remainder R. In direct method, AMD ordering scheme can minimize the storage and the number of floating-point operations.

7.2.1 Effect of AMD Ordering on Structure of Matrix

I have made some investigation on how the AMD ordering scheme deals with the large sparse coefficient matrix. We take the NG 953 bus system case as the sample.

In the figure, the dot denotes the location of nonzero elements in the Jacobian matrix, and the equation on the bottom of the figures refers to the amount of nonzero elements.



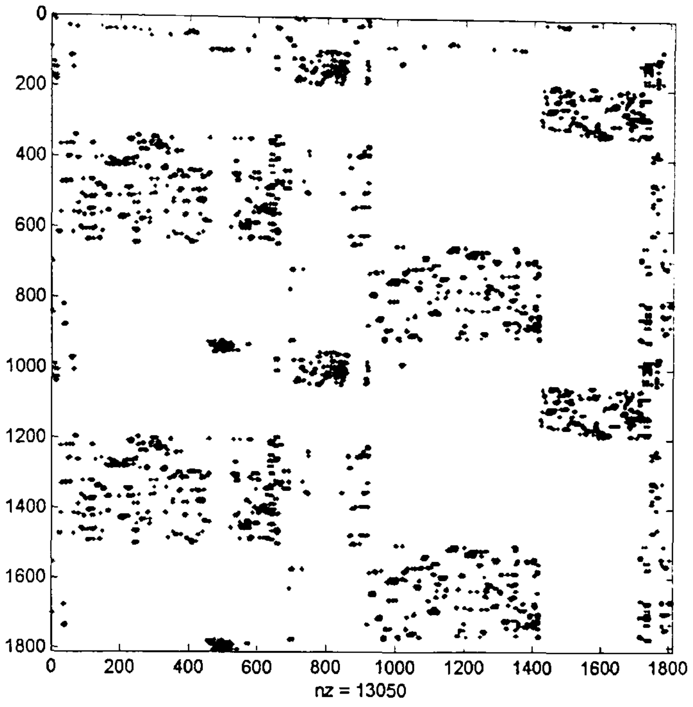
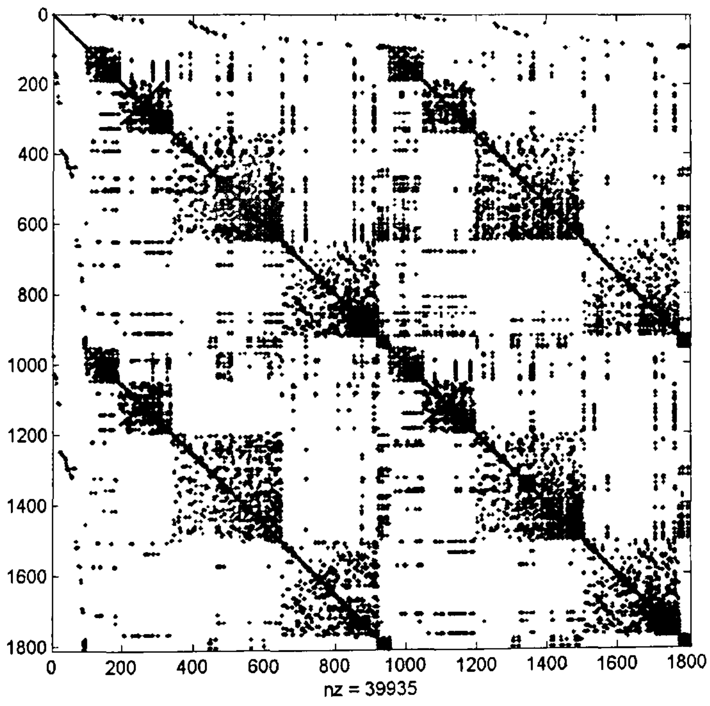


Figure 7-1-a: Sparsity pattern of unordered Jacobian matrix of NG 953 bus system

Figure 7-1-b: Sparsity pattern of reordered Jacobian matrix of NG 953 bus system



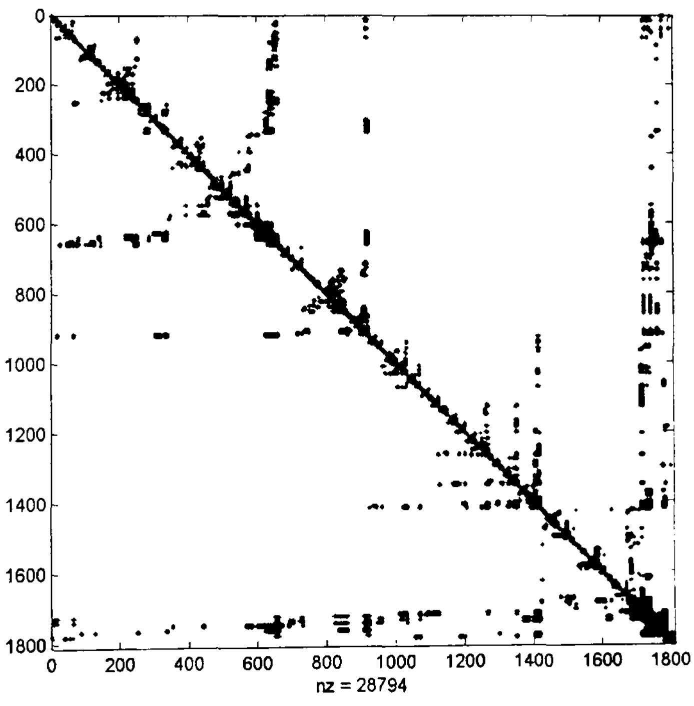


Figure 7-2-a: Sparsity pattern of ILU pre-conditioner of unordered Jacobian matrix

Figure 7-2-b: Sparsity pattern of ILU pre-conditioner of reordered Jacobian matrix

Figures 7-1-a and 7-1-b illustrate the effect of the AMD ordering scheme on the structure of the Jacobian matrix. We observe that the number of nonzero elements is equal in both of the matrices from the figures 7-1-a and 7-1-b, which means the AMD ordering scheme is able to preserve the sparsity of the Jacobian matrix, the AMD just relocates the nonzero elements in the matrix. Figures 7-2-a and 7-2-b were obtained by applying ILU factorization to the unordered Jacobian matrix and the AMD reordered Jacobian matrix respectively, which illustrates the effect on the structure of preconditioned Jacobian matrix.

The AMD ordering scheme makes the nonzero elements of preconditioned Jacobian matrix higher density and closer locating to the main diagonal in Figure 7-2-b.

7.2.2 Effect of AMD Ordering on Eigenvalue of Matrix

In this section, we tested how the AMD ordering scheme affected the eigenvalue spectrum, which in turn has an impact on convergence rates. We take IEEE 118 bus system case as a sample to illustrate the impact.

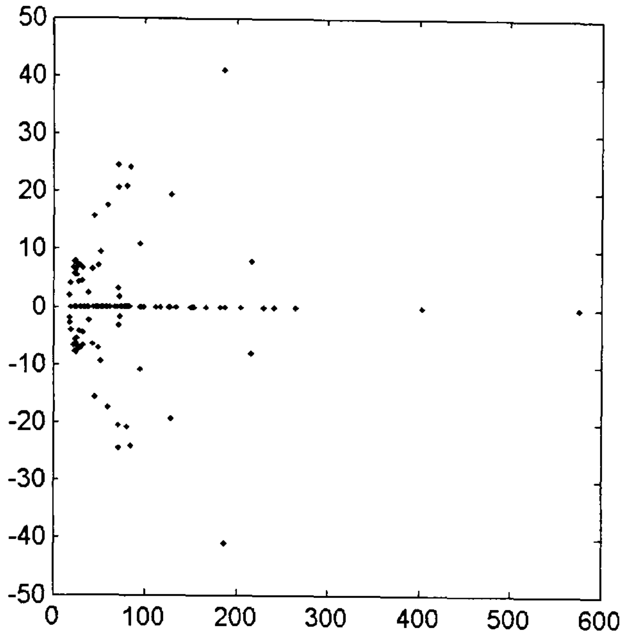


Figure 7-3-a: Distribution of eigenvalues of unordered Jacobian matrix

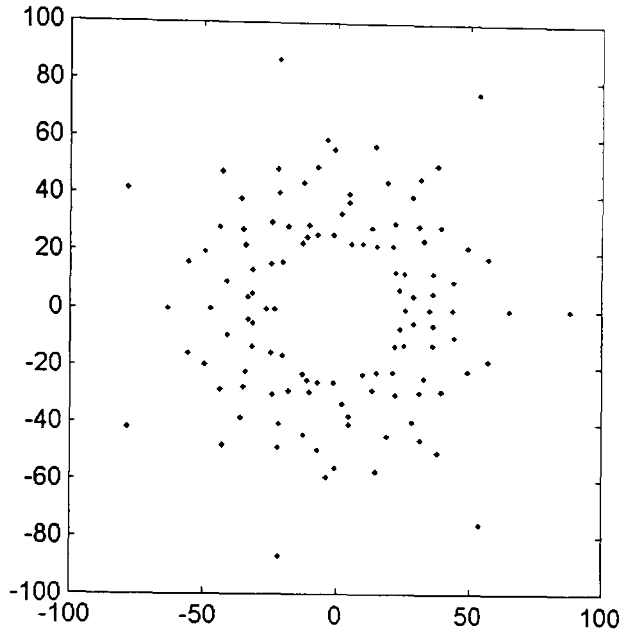


Figure 7-3-b: Distribution of eigenvalues of reordered Jacobian matrix

The dots in the figures refer to the location of eigenvalues of the Jacobian matrix. Figures 7-3-a and 7-3-b illustrate the distribution of eigenvalues. From a comparison of both figures, it is obvious that the AMD reordered the Jacobian matrix produces clustering and clumping of the eigenvalues, and the effect can be observed by comparing the maximal eigenvalues of the unordered and reordered matrix. In other words, the AMD ordering scheme can change the distribution of eigenvalues of the Jacobian matrix, and group the eigenvalues together.

As a result, the convergence rate of iterative methods with the pre-conditioner using the AMD reordered ILU factorization will be reduced comparing with that without the AMD reordered.

7.2.3 Effect of AMD Ordering on Computing Time

Test System	GMRES		BiCG		BiCGStab	
	unordered	reordered	unordered	reordered	unordered	reordered
IEEE 30	0.0383	0.0137	0.0287	0.0178	0.0267	0.0167
IEEE118	0.0780	0.0297	0.0778	0.0296	0.0661	0.0266
IEEE300	0.8119	0.0984	0.3735	0.0987	0.7364	0.0823
POL2736	14.5408	0.7710	16.7340	0.7845	14.8688	0.6594
NG61	0.0424	0.0280	0.0455	0.0272	0.0416	0.0252
NG953	1.9310	0.2747	2.2233	0.2827	1.9157	0.2371

Table 7-1: Computing time comparison between unordered and reordered pre-conditioner used in iterative solvers

Time recorded is in second. The “unordered” and “reordered” refer to the time of the iterative methods that using unordered ILU factorization pre-conditioner and using the AMD reordered ILU factorization pre-conditioner respectively.

In this section, I compared the computational time of power flow computation for each test case. From the results Table 7-1, it is clearing apparent that the iterative methods using the AMD reordered pre-conditioner are always faster than those without using that. The large bus system cases obtain more benefits from the AMD ordering scheme than small systems, especially in the Poland 2736 test case.

7.3 Comparison and Analysis of Linear Solvers

The direct method investigated in this research is the Gaussian Elimination method. A direct method solves the linear equations in a finite number of steps and may require extensive computational storage and arithmetic processing. Direct methods have been improved over the years through the use of efficient ordering techniques and sparse programming. However, the computation time of a direct method can still be the limiting factor in cases involving very large systems of equations or numerous repeated solutions.

In the Newton-Raphson power flow, the Jacobian matrix is an asymmetric, possibly indefinite matrix. For asymmetric linear systems, there are a number of iterative methods that can be employed, the GMRES, BiCG and BiCGStab methods is of interest in this research work.

In this section, I compared the direct method with iterative methods in computational time and accuracy.

Test System	Max P&Q Mismatch (1e-11)			
	Direct	GMRES	BiCG	BiCGStab
IEEE30	95.70	95.70	95.70	95.70
IEEE118	0.15188	0.15060	0.15083	0.15188
IEEE300	0.13754	0.13846	0.13723	0.13781
POL2736	2.0522	2.8411	3.7811	2.0561
NG61	30.158	30.157	30.161	30.159
NG953	0.18758	0.18723	0.20464	0.19369

Table 7-2: Comparison of direct method with different iterative methods in accuracy

In Table 7-2, "Max P&Q Mismatch" refers to the maximum value of the active and reactive power mismatches. All the results obtained under the same accuracy requirement.

Test system	Computational Time (s)				Difference (%)		
	Direct	GMRES	BiCG	BiCGStab	GMRES	BiCG	BiCGStab
IEEE30	0.0156	0.0173	0.0178	0.0167	9.82	12.36	6.59
IEEE118	0.0257	0.0297	0.0296	0.0266	13.47	13.18	3.38
IEEE300	0.0795	0.0984	0.0987	0.0823	19.21	19.45	3.40
POL2736	0.6077	0.7710	0.7845	0.6594	21.18	22.54	7.84
NG61	0.0231	0.0280	0.0272	0.0252	17.50	15.07	8.33
NG953	0.2175	0.2747	0.2827	0.2371	20.82	23.06	8.27

Table 7-3: Comparison of direct method with different iterative methods in computational time

In Table 7-3, “Difference” refers to the percentage slow down of the computation time of the iterative methods compared to the direct method.

In order to obtain meaningful comparisons between direct and iterative methods, it is necessary to ensure the same general testing conditions for all of the methods. The AMD ordering scheme and ILU matrix factorization algorithms were used as pre-conditioner in the direct and iterative methods. From the results we observe that the iterative methods are comparable in computational time with the direct method when solving large sparse, asymmetrical, indefinite matrices to the same level of accuracy. The AMD ordering scheme benefits the iterative methods as well as the direct method. Iterative methods using the AMD reordered ILU factorization matrix as a pre-conditioner effectively reduce the number of iterations and speed up the convergence procedure.

It is well known that the parallelization and factorization are easily applicable to iterative methods when solving large systems of linear equations, but direct methods are known to be hard to parallelize and factorize. So it may be possible that the iterative methods still have more potential to speed up computation than direct methods.

7.4 Summary

The AMD ordering scheme for ILU factorization algorithm is a very good pre-conditioner for implementation of iterative linear solvers, which clusters and clumps the eigenvalue, reduces the convergence rate, and saves computing time efficiently. The effort of new pre-conditioner makes iterative methods comparable to the conventional direct method in computational time even for the small bus systems.

The AMD reordered ILU pre-conditioner secures all the iterative methods convergence successfully, and provides a feasible environment to implement iterative methods.

Chapter 8: Conclusions and Future Work

This chapter summarizes the conclusions of the whole thesis, and presents the conclusions from the comparisons between CCPF, ECPF and GCPF, and the comparisons between direct method and iterative linear solvers used in power flow computation. All of the comparing experiments are implemented under the same condition for each case. We also suggest some further work for my project in the future.

8.1 Conclusions

In this thesis, I have made plenty of experiments to test the proposed GCPF method. I also compare the results of GCPF with CCPF and ECPF methods in the numbers of continuation steps and computing time. In order to illustrate the efficiency of GCPF, the experiments are conducted on different size network cases: IEEE 9, IEEE 30, IEEE 118, National Grid (NG) UK 61 and 953 bus systems.

It is well known that solving large sparse linear equations plays important role in Newton-Raphson power flow computation, so the different linear solvers have also been investigated in this thesis, the linear solvers include direct method and iterative methods. The direct method is the Gaussian Elimination method, the iterative methods are GMRES, BiCG and BiCGStab methods. The iterative methods for solving large sparse sets of linear equations can naturally profit from parallelization and factorization. I also investigate the effect of a good pre-conditioner on convergence rate and computing time of iterative solvers. In this part of research work, I choose some large power systems to implement the experiments, which are IEEE 300, Poland 2736 and NG 953 bus systems.

8.1.1 Novel Q-limit Guided Continuation Power Flow

This thesis presents a new algorithm for computing the voltage collapse point, which is the Novel Q-limit Guided Continuation Power Flow method. The GCPF method is based on the standard CPF method, and combines Q-limit breaking indices prediction and saddle-node bifurcation point prediction together. The predicted Q-limit index parameters are used to guide the continuation step to a Q-limit breaking point in the fastest way possible. From our experiments, it is obvious that GCPF method works well. It has been found to be much faster than the CCPF and ECPF methods for the IEEE 9, 30, 118, NG 61 and NG 953 bus system case studies when determining a reliable voltage collapse index in terms of maximal load margin. The GCPF method gives the best performance in cases that do not have too many generators reaching their reactive power limits in separate continuation steps.

In order to make a meaningful comparisons between CCPF, ECPF and GCPF, we not only test simple case like IEEE 9 bus system, but also the large case like IEEE 118 bus system, a realistic power system NG 61 and NG 953 bus system are also tested. Different case studies have their own structures and feathers, which effect the implementation of GCPF method effectively. The conclusions are properly different in terms of different size of case studies, which have been described in detail for each case.

IEEE 9 Bus System Case

During the computation for saddle-node bifurcation point, there is only one Q-limit breaking point that occurred, which is very close to the bifurcation point. In this case, the IEEE 9 bus system does not provide a good condition to implement the proposed GCPF method, because the Q-limit breaking point is quite far from the initial solution and the Lagrange polynomial interpolation formula is hard to predict the location of Q-limit index accurately, this would result in GCPF taking many unnecessary continuation steps to reach the Q-limit breaking point. However, the GCPF presents much better performance than CCPF and ECPF in terms of computational steps required and computing time, but it is not the optimal results we expected.

IEEE 30 and 118 Bus System Cases

In these two case studies, the data set of IEEE 30 and 118 bus systems provide a very good condition to implement the proposed GCPF method. The results obtained are what we expected, the continuation steps have been efficiently guided by the Q-limit indices. Therefore, GCPF method reduces the number of steps required significantly and saves substantial computing time. With comparing with CCPF and ECPF methods, GCPF shows the best performance in the number of required computational steps and computational time within the same accuracy requirement when determining a reliable voltage collapse index in terms of maximum load margin.

From the results of IEEE 30 and 118 bus systems, the conclusions can be obtained are that the performance of GCPF method strongly depends on the distribution of Q-limit breaking points. GCPF will present the optimal results when Q-limit breaking points locate on the P-V curve sparsly and evenly, this situation can generate the serial valid Q-limit indices to continuously guide the current step to reach the Q-limit breaking point or the critical point in a fast way possible.

NG 61 and NG 953 Bus System Cases

These two bus systems are realistic power systems, they are very complicated systems. In these case studies, GCPF is still faster than CCPF and ECPF when determining the maximum load margin, but GCPF did not present the optimal results in some cases. In multiple load bus variation case of NG 61 and area 5 of NG 953, GCPF still works much more efficient than CCPF, but is comparable to the ECPF method, the reason is that there are too many Q-limit breaking points occurred during the computation process. Under this condition the Q-limit breaking points are too many and too close to each other, it is hard for GCPF to reduce the number of computational steps, because the data collection of Q-limit predictor requires at least three solutions for each procedure in the GCPF algorithm. So, the valid Q-limit index is not able to save continuation steps effectively. In single load bus variation case of NG 61, we observe another advantage of GCPF comparing with ECPF, which is that GCPF is able to secure the accuracy requirement for every single calculation of Q-limit breaking point during the whole computation process.

From the analysis on results of NG 61 and NG 953 bus systems, another conclusion can be obtained is that the high ratio of Q-limit breaking point to total number of continuation steps is not a good condition to implement the proposed GCPF method, in other words, GCPF method can not present the best performance in cases, which has too many generators reaching their reactive power limit in separate continuation steps. In any case, GCPF does speed up the conventional CPF method.

Summary

The proposed GCPF method works very well, and did improve the standard CPF method. I summarize the feathers of the novel GCPF algorithm in details by advantages and disadvantages.

Advantages:

- The proposed GCPF is generally computationally faster than the standard CPF and ECPF methods;
- GCPF method can compute the voltage collapse index for each case under the required accuracy quickly;
- GCPF method can secure the accuracy requirement of every Q-limit breaking point computation in the whole computation process;
- GCPF takes account of reactive power output of generators when determining the maximum load margin.

Disadvantages:

- The performance of the GCPF method is affected by the condition of system data set;
- The performance of the GCPF method strongly depends on the number and distribution of Q-limit breaking points.

8.1.2 Preconditioned Iterative Linear Solvers

In order to obtain meaningful comparisons between direct and iterative methods, it is necessary to ensure the same general testing conditions for all of the methods. The column approximate minimum degree (AMD) algorithm and incomplete LU (ILU) matrix factorization are used in the direct and iterative methods. From the results I observe that the iterative linear solvers are comparable in computational time to the direct method when solving large sparse, asymmetrical, indefinite matrices to the same level of accuracy and even for small networks such as IEEE 30 and NG 61 bus systems. The AMD ordering scheme benefits the iterative methods as well as the direct method. Iterative methods using the AMD reordered ILU factorization as pre-conditioners reduce the number of iterations and speed up the convergence rate.

It is well known that the parallelization and factorization are easily applicable to iterative methods when solving large systems of linear equations, but direct methods are known to be hard to parallelize and factorize. So it may be possible that the iterative methods still have more potential to speed up computation than direct methods, which remains to be explored in the future.

How does the AMD ordering scheme impact on structure of Jacobian matrix, eigenvalue spectrum, convergence rate and computing time are fully discussed as well, we summarizes the conclusions:

- The AMD ordering scheme make the element location higher density closer to the main diagonal of the matrix;
- The AMD ordering scheme produces both clustering and clumping the eigenvalue spectrum;
- The AMD ordering scheme reduce the number of iterations and convergence rate of preconditioned iterative methods;
- The application of AMD ordering scheme for pre-conditioners based on ILU factorization can speed up the computation of iterative linear solvers effectively.

This thesis presents the results obtained from the comparative studies, which designed to evaluate the performances of direct method against iterative solvers when solving linear equations in Newton-Raphson power flow calculations. The general conclusion is that direct methods are still faster, though iterative methods are easier to implement. Also iterative methods have more potential to speed up because iterative methods would benefit more from parallelization and factorization than direct method.

The AMD ordering scheme for pre-conditioners based on ILU factorization is a key factor to secure successful convergence for all iterative methods. AMD ordering scheme is able to accelerate and lead all iterative solvers to successful convergence, which is applied to a wide range of networks such as Poland 2736 buses.

The BiCGStab with AMD ordered ILU pre-conditioner set presents the best performance comparing with other iterative methods from the numerical results. The average percentage slower down than direct method in computational time is just 6.30%, in other words, this set of algorithm is strong comparable to the direct solution method.

The GMRES with AMD ordered ILU pre-conditioner set algorithm shows good robustness, however, it is slower than BiCGStab. The BiCG with AMD ordered ILU pre-conditioner set algorithm shows smooth convergence properties even for large bus systems. Although being the slowest one, this method is still very appealing.

The AMD ordered ILU factorization pre-conditioner shows almost perfect parallelism, and can be applied easily in all iterative methods. A parallel implementation will reduce the total computation time significantly.

8.2 Future work

The step size in CCPF and GCPF algorithms is selected as fixed but optimized to ensure that both of the methods have minimal continuation steps for the requirement of accuracy. The same step size is used in the continuation steps of GCPF method except the Q-limit index guiding steps, which is not optimized. It is possible to use a variable step size strategy to improve GCPF method further but this remains to be explored.

The proposed GCPF algorithm can be further optimized in implementation or by using other related index predicting methods instead of Lagrange polynomial interpolation technique.

In further research, I will investigate whether we can combine the GCPF with nonlinear predictor instead of tangent vector or rule based step size control together. In addition, I will try to improve the prediction accuracy of Q-limit breaking points further.

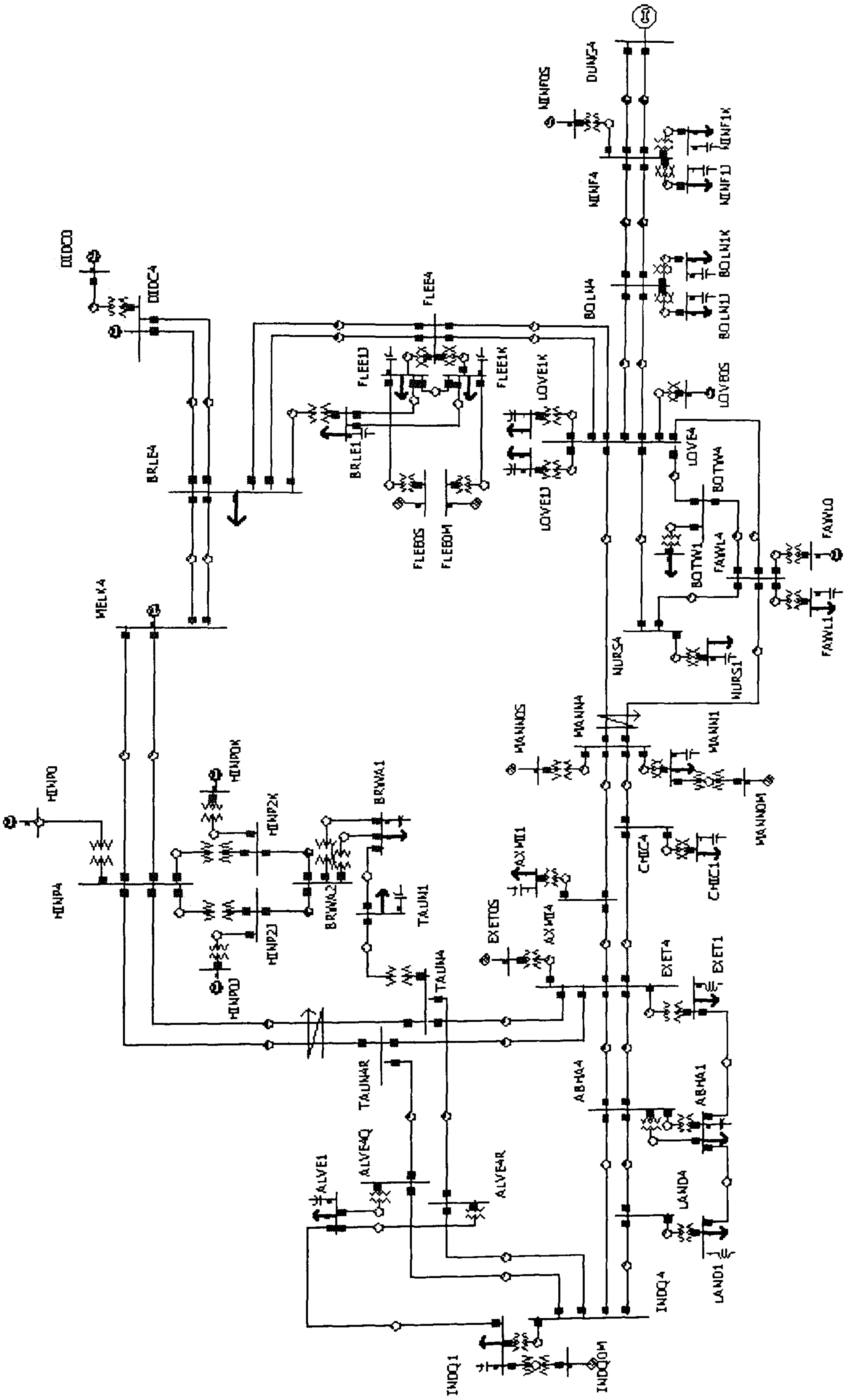
In linear solvers, we will investigate the parallelization and factorization of iterative methods, matrix ordering schemes and the application of pre-conditioners to speed up iterative linear solvers in power flow calculations. From the comparison and analysis, it can be foreseen that the use of more sophisticated pre-conditioners, a deeper understanding of the characteristics of these methods applied to the power flow and the use of the parallel CPU environments will further improve performance and robustness.

In addition the benefits of the faster GCPF algorithm will be further investigated when determine the margins to voltage collapse for large numbers of network contingencies.

OLTCs (on load tap changers) are voltage devices, which need to be investigated with the proposed GCPF.

Appendix A

Illustration of NG 61 Bus System



Appendix B

Matlab Programme Code of Predictor

```
function [VO, AO] = predictor(Ybus, Sbus, V, A, n, h, ref, pv, pq)


j = sqrt(-1);
Va = angle(V);
Vm = abs(V);
h = h;
b = zeros(size(Sbus));
b(n) = Sbus(n);
B = [     real(b([pv; pq]));
            imag(b(pq))     ];


npv = length(pv);
npq = length(pq);
j1 = 1;         j2 = npv;
j3 = j2 + 1;  j4 = j2 + npq;
j5 = j4 + 1;  j6 = j4 + npq;


[dSbus_dVm, dSbus_dVa] = dSbus_dV(Ybus, V);


    temp = real(dSbus_dVa(:, [pv; pq]))';
    j11 = temp(:, [pv; pq])';
    temp = real(dSbus_dVm(:, pq))';
    j12 = temp(:, [pv; pq])';
    temp = imag(dSbus_dVa(:, [pv; pq]))';
    j21 = temp(:, pq)';
```

```
  temp = imag(dSbus_dVm(:, pq))';
  j22 = temp(:, pq)';


  J = [     j11 j12;
      j21 j22;   ];


  ds = (J \ B);


  da = sqrt((1 + sum(ds .* ds))^(-1));


  ds = da * ds;
  Va(pv) = Va(pv) + h * ds(j1:j2);
 Va(pq) = Va(pq) + h * ds(j3:j4);
 Vm(pq) = Vm(pq) + h * ds(j5:j6);


 VO = Vm .* exp(j * Va);
 AO = A + h * da;


 return;
```

Appendix C

Matlab Programme Code of Corrector

```
function [V, A, converged, i] = corrector(Ybus, Sbus, VO, V, AO, A, n, h, ref, pv, pq,
mpopt)


%% default arguments
if nargin < 7
    mpopt = mpoption;
```

```
end

%% options
tol     = mpopt(2);
max_it  = mpopt(3);
verbose = mpopt(31);

%% initialize
j = sqrt(-1);
converged = 0;
i = 0;
v = V;
V = VO;
a = A;
A = AO;
Va = angle(V);
Vm = abs(V);
va = angle(v);
vm = abs(v);
%%ds = sqrt(sum((Va([pv;pq]) - va([pv;pq])) .^2) + sum((Vm(pq) - vm(pq)) .^2) + (A
- a)^2);
ds = h;

%% set up indexing for updating V
npv = length(pv);
npq = length(pq);
j1 = 1;         j2 = npv;          %% j1:j2 - V angle of pv buses
j3 = j2 + 1;    j4 = j2 + npq;     %% j3:j4 - V angle of pq buses
j5 = j4 + 1;    j6 = j4 + npq;     %% j5:j6 - V mag of pq buses
j7 = j6 + 1;
```

```
%% evaluate F(x0)
b = zeros(size(Sbus));
b(n) = Sbus(n);
mis = V .* conj(Ybus * V) - A * b - Sbus;
Mis = sum((Va([pv;pq]) - va([pv;pq])) .^2) + sum((Vm(pq) - vm(pq)) .^2) + (A - a)^2 -
ds^2;
F = [     real(mis([pv; pq]));
          imag(mis(pq));
           Mis       ];

%% check tolerance
normF = norm(F, inf);
if verbose > 1
     fprintf('\n it       max P & Q mismatch (p.u.)');
     fprintf('\n----   ---------------------------');
     fprintf('\n%3d               %10.3e', i, normF);
end
if normF < tol
     converged = 1;
     if verbose > 1
          fprintf('\nConverged!\n');
     end
end


%% do Newton iterations
while (~converged & i < max_it)
     %% update iteration counter
     i = i + 1;
```

```
%% evaluate Jacobian
[dSbus_dVm, dSbus_dVa] = dSbus_dV(Ybus, V);

temp = real(dSbus_dVa(:, [pv; pq]))';
j11 = temp(:, [pv; pq])';
temp = real(dSbus_dVm(:, pq))';
j12 = temp(:, [pv; pq])';
 j13 = - real(b([pv; pq]));
temp = imag(dSbus_dVa(:, [pv; pq]))';
j21 = temp(:, pq)';
temp = imag(dSbus_dVm(:, pq))';
j22 = temp(:, pq)';
j23 = - imag(b(pq));

 j31 = 2 * (Va([pv;pq]) - va([pv;pq]))';
 j32 = 2 * (Vm(pq) - vm(pq))';
 j33 = 2 * (A - a);

J = [      j11 j12 j13;
           j21 j22 j23;
           j31 j32 j33;      ];

%% compute update step
dx = -(J \ F);

%% update voltage
Va(pv) = Va(pv) + dx(j1:j2);
Va(pq) = Va(pq) + dx(j3:j4);
Vm(pq) = Vm(pq) + dx(j5:j6);
```

```
      A = A + dx(j7);
      V = Vm .* exp(j * Va);


      %% evalute F(x)
  mis = V .* conj(Ybus * V) - A * b - Sbus;
  Mis = sum((Va([pv;pq]) - va([pv;pq])) .^2) + sum((Vm(pq) - vm(pq)) .^2) + (A -
        a)^2 - ds^2;


      F = [    real(mis(pv));
               real(mis(pq));
               imag(mis(pq));
               Mis    ];


      %% check for convergence
      normF = norm(F, inf);
      if verbose > 1
      fprintf('\n%3d             %10.3e', i, normF);
      end
      if normF < tol
          converged = 1;
          if verbose
            %fprintf('\nNewton''s method power flow converged in %d iterations.\n', i);
          end
      end
  end


  if verbose
      if ~converged
          fprintf('\nNewton''s method power did not converge in %d iterations.\n', i);
      end
```

```
end

return;
```

Appendix D

Matlab Programme Code of Q-limit Predictor

```
function [AO] = Alimpredictor(Qg, Al, Qlim)


AO = [];

BO = [];

for i = 1 : length(Qg(:, 1))


    k11 = Qg(i, :) .^ 2;

    k12 = Qg(i, :);

    k13 = ones(1, 3);


    J = [k11', k12', k13'];


    abc = J \ Al';

    a = abc(1);

    b = abc(2);

    c = abc(3);


    Q = - b / (2 * a);

    Qmax = Qlim(i, 1);

    Qmin = Qlim(i, 2);


    if Q >= Qmax

        AO(i) = a * Qmax ^ 2 + b * Qmax + c;
        elseif Q <= Qmin
                AO(i) = a * Qmin ^ 2 + b * Qmin + c;


        else
                BO(i) = (4 * a * c - b * b) / (4 * a);
        end
end


na = find(AO(:) ~= 0);
nb = find(BO(:) ~= 0);


if ~isempty(AO(na))
        AO = min(AO(na));
else
        AO = min(BO(nb));
end
```

References

[1] H.-D. Chiang, Rene J.-J, “Toward a Practical Performance Index for Predicting Voltage Collapse in Electric Power System.” *IEEE Trans. Power Syst.*, vol. 10, no. 2, pp. 584-592, May 1995.

[2] A. Kurita and T. Sakurai, “The power system failure on July 23, 1987 in Tokyo,” *IEEE Proceedings of the 27th Conference on Decision and Control*, Austin, TX, pp. 2093-2097, Dec 1988.

[3] Y. Mansour, “Voltage Stability of Power Systems: Concepts, Analytical Tools and Industry Experience,” *IEEE Power Systems Engineering Committee*, 1991.

[4] I. Dobson, H. Glavitsch, C.-C. Liu, Y. Tamura and K. Vu, “Voltage Collapse in Power Systems” *IEEE*, pp. 40-45, May 1992.

[5] N. Flatabø, R. Ognedal and T. Carlsen, "Voltage stability condition in a power transmission system calculated by sensitivity methods", *IEEE Trans. Power Systems*, Vol. 5, No. 4, p.1286-1293, Nov. 1990.

[6] J.Q. Zhao, H.D. Chiang and H. Li, "Enhanced look-ahead load margin estimation for voltage security assessment", *Electrical Power and Energy Systems*, Vol.26, pp. 431-438, 2004.

[7] I. Dobson, S. Greene, R. Rajaraman, C. L. DeMarco, F. L. Alvarado, M. Glavic. J.F. Zhang, R. Zimmerman, “Electric Power Transfer Capability: Concepts, Applications, Sensitivity and Uncertainty,” *PSERC Publication* 01-34 November 2001.

[8] G. A. Taylor, S. Phichaisawat, M.R. Irving and Y.-H. Song, "Voltage Security and Reactive Power Management," *IMA Journal of Management Mathematics*. Vol. 15, Pgs. 369-386, 2004.

[9] H.-D. Chiang, H. Li, "CPFLOW for Power Tracer and Voltage Monitoring, Voltage Collapse Margin Monitor Final Report," *PSERC Publication 01-02*, May 2002.

[10] H.-D. Chiang, A. J. Flueck, K. S. Shah, N. Balu, "CPFLOW: A Practical Tool for Tracing Power System Steady-State Stationary Behaviour Due to Load and Generation Variations," *IEEE Transactions on Power Systems*, Vol. 10, No. 2, pp. 623-630, May 1995.

[11] V. Ajjarapu, C. Christy, "The Continuation Power Flow: A Tool For Steady State Voltage Stability Analysis," *Transactions on Power Systems*, Vol. 7, No. 1, pp. 416-423, February 1992.

[12] N. Yorino, H.-Q Li, and H. Sasaki, "A Predictor/Corrector Scheme for Obtaining Q-Limit Points for Power Flow Studies," *IEEE Transactions on Power Systems*, VOL. 20, NO. 1, pp. 130-137, Feb 2005.

[13] F. W. Mohn and A. C. Zambroni de Souza, "Tracing PV and QV Curves With the Help of a CRIC Continuation Method," *IEEE Transactions on Power Systems*, Vol. 21, No. 3, pp. 1115-1122, August 2006.

[14] C. A. Canizares, A. Z. de Souza, and V. H. Quintana, "Comparison of performance indices for detection of proximity to voltage collapse," *IEEE Trans. Power Syst.*, vol. 11,pp. 1441-1450, Aug. 1996.

[15] H. D. Chiang and R. Jean-Jumeau, “A more efficient formulation for computation of the maximum loading points in electrical power systems,” *IEEE Trans. Power Syst.*, vol. 10, pp. 635-646, May 1995.

[16] G. C. Ejebe, G. D. Irisarri, S. Mokhtari, O. Obadina, P. Ristanovic, and J. Tong, “Methods for contingency screening and ranking for voltage stability analysis of power systems,” *IEEE Trans. Power Syst.*, vol. 11, pp. 350-356, Feb. 1996.

[17] A. O. Ekwue, H. B. Wan, D. T. Y. Cheng, and Y. H. Song, “voltage stability analysis on the NGC systems,” *Elec. Power Syst. Res.*, vol.47, pp. 173-180, 1998.

[18] S. Greene and I. Dobson, “Voltage Collapse Margin Sensitivity Methods Applied to the Power System of Southwest England,” *NGC*, 1998.

[19] S. Greene, I. Dobson, and F. L. Alvarado, “Sensitivity of the loading margin to voltage collapse with respect to arbitrary parameters,” *IEEE Trans. Power Syst.*, vol. 12, pp. 262-272, Feb. 1997.

[20] S. Greene, I. Dobson, and F. L. Alvarado, “Contingency ranking for voltage collapse via sensitivities from a single nose curve,” *IEEE Trans. Power Syst.*, vol. 14, pp. 232-240, Feb. 1999.

[21] S. Greene, I. Dobson, and F. L. Alvarado, “Sensitivity of transfer capability margins with a fast formula,” *IEEE Trans. Power Syst.*, vol. 17, pp. 34-40, Feb. 2002.

[22] R. Jean-Jumeau, H.-D Chiang, and R. T. Thomas, “Efficient computational methods for a practical performance index and the exact voltage collapse point

in electrical power system," in *proc.31st Conf. Decision Contr.*, Tucson, AZ, USA, 1992.

[23] P. A. Lof, T. Smed, G. Anderson, and D. J. Hill, "Fast calculation of a voltage stability index," *IEEE Trans. Power Syst.*, vol. 7, pp. 54-64, Feb 1992.

[24] R. Seydel, "Numerical computation of branch points in nonlinear equations," *Numer. Math.*, vol. 33, pp. 339-352, 1979.

[25] R. Seydel, "On detecting stationary bifurcations," *Int. J. Bifurcations and Chaos*, vol. 1, no. 2, pp. 335-337, 1991.

[26] R. Seydel, "Practical Bifurcation and Stability Analysis," *Equilibrium to Chaos.* New York: Springer-Verlag, 1994.

[27] R. Seydel, "On a class of bifurcation test functions," *Chaos, Solitons and Fractals*, vol. 8, no. 6, pp. 851-855, 1997.

[28] R. Seydel, "Assessing voltage collapse," Proc. Risk Analysis: *Opening the Process Conf.*, Paris, France, Oct. 11-14, 1998.

[29] H. Mori, Member IEEE, and T. Kojima, "Hybrid Continuation Power Flow with Linear-Nonlinear Predictor," *International Conference On Power System Technology - POWERCON 2004 Singapore*, pp. 969-974, 21-26 Nov 2004.

[30] H. Mori and S. Yamada, "Continuation Power Flow with the Nonlinear Predictor of Lagrange's Interpolation Formula," *Proc. Of IEEE PES Transmission and Distribution Conference and Exhibition 2002. Asia Pacific*, Vol. 2, pp. 1133-1138, Oct 2002.

[31] V. Ajjarapu, N. Jain, Z. Yu, and S. Barttula, "Recent developments to the continuation power flow," in *proc. North Amer. Power symp.*, Washington D.C., Oct. 1993, pp. 205-412.

[32] A, J. Flueck and J. R. Dondeli, "A new continuation power flow tool for investigating the nonlinear effects of transmission branch parameter variations," *IEEE Trans. On Power Systems*, vol. 15, no. 1, pp. 223-227, Feb 2000.

[33] M. R. Hestenes, E. Stiefel, "Methods of conjugate gradients for solving linear systems," *J. Res. National bureau of standards*, vol. 49, pp. 409-436, 1952.

[34] F. D. Galiana, H. Javidi, S. McFee, "On The Application of A Pre-conditioned Conjugate Gradient Algorithm to Power Network Analysis", *IEEE Transactions on Power Systems*, Vol. 9, No. 2, pp. 629-635, May 1994.

[35] H. Mori, H. Tanaka, and J. Kanno, "A Pre-conditioned Fast Decoupled Power Low Method For Contingency Screening", *IEEE Transactions on Power Systems*, Vol. 11, pp. 357-363, Feb 1996.

[36] A. Semlyen, "Fundamental concepts of a Krylov subspace power flow methodology," *IEEE PES Summer Meeting*, July 23-27, 1995.

[37] R. Bacher and E. Bullinger, "Application of Non-stationary Iterative Methods to an Exact Newton-Raphson Solution Process for Power Flow Equations," *12th Power Systems Computation Conference*, pp. 453-459, Aug 1996.

[38] Y. Chen and C. Shen, "A Jacobian-Free Newton-GMRES(m) Method with Adaptive Preconditioner and Its Application for Power Flow Calculations," *IEEE Trans. On Power Systems*, vol. 21, no. 3, Aug 2006.

[39] A.B. Alves, E.N. Asada and A. Monticelli, "Critical Evaluation of Direct and Iterative Methods for Solving Ax = b Systems in Power Flow Calculations and Contingency Analysis", IEEE Transactions on Power Systems, Vol.14, No.2, pp.702-708, May 1999.

[40] B. Stott, O. Alsac and A. Monticelli, "Security Analysis and Optimization," *Invited Paper. Proc. Of the IEEE*, vol. 75, no. 12, pp. 1623-1644, Dec 1987.

[41] M. K. Enns, J. J. Quada and B. Sacckett, "Fast Linear Contingency Analysis," *IEEE Trans. On PAS*, vol. 101, no. 4, pp. 783-791, Apr 1982.

[42] O. Alsac, B. Stott and W. F. Tinney, "Sparsisty-Oriented Compensation Methods for Modified Network Solutions," *IEEE Trans. On PAS*, vol. 102, pp. 1050-1060, May 1983.

[43] W. F. Tinney, V. Brandwajn and S. M. Chan, "Sparse Vector Methods," *IEEE Trans. On PAS*, vol. 104, no. 2, pp. 295-301, Feb 1983.

[44] S. M. Chan and V. Brandwajn, " Partial Matrix Refactorization," *IEEE Trans. On Power Systems*, vol. PWRS-1, no. 1, pp. 193-200, Feb 1986.

[45] I. C. Decker, D. M. Falcao and E. Kaszkurewicz, "Conjugate Gradient Methods for Power System Dynamic Simulation on Parallel Computers," *IEEE PES Summer Meeting*, July 1995.

[46] A. Gupta, V. Kumar and A. Sameh, "Performance and Scalability of Preconditioned Conjugate Gradient Methods on Parallel Computers," *IEEE Trans. On Parallel and Distributed Systems*, vol. 6, no. 5, May 1995.

[47] H. Dag, Member, IEEE, A. Semlyen, "A New Preconditioned Conjugate Gradient Power Flow" *IEEE Transactions on Power Systems*, Vol. 18, No. 4, pp. 1248-1255, Nov 2003.

[48] Y. Saad and M. H. Schultz, "GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems," *SIAM J. Sci. Statist. Comput*, Vol. 7, No. 3, July 1986, pp. 856-869.

[49] R. Barrett, M. Berry, T. F. Chan, J. Demmel and J. Donato, "Templates for the solution of linear systems: Building Blocks for Iterative Methods," *2nd Edition, SLAM*, Philadelphia, PA, 1994.

[50] H. Su, Yi. Zhang, Yu. Zhang and J. Man, "A Compressed BiCGStab Algorithm for Power and Ground Network Analysis," *IEEE*, pp. 1233-1236, 2007.

[51] P. C. Zhu, G. Taylor and M. Irving, "A Novel Q-limit Guided Continuation Power Flow Method," *IEEE PES General Meeting Proceeding*, Pittsburgh USA, July 2008.

[52] C. A. Canizares, "Voltage stability indices," in Voltage Stability Assessment. Procedures and Guides: *IEEE/Power Eng. Soc.* PSS Subcommittee Special Publication, 2002, ch. 4.

[53] K. Chen, A. Hussein, M. E. Bradley, and H.-B Wan, " A Performance-Index Guided Continuation Method for Fast Computation of Saddle-Node Bifurcation

in Power System," *IEEE Transactions on Power Systems*, Vol. 18, No. 2, pp. 753-760, May 2003.

[54] I. A. Hiskens and B. B. Chakrabati, "Direct calculation of reactive power limit points," *Int. J. Elect. Power and Energy Syst*, Vol. 18, No 2, PP. 121-129, 1996.

[55] N. Kockler, "Numerical Methods and Scientific Computing," *Oxford University Press Inc.*, New York, USA, 1994.

[56] Timothy A. Davis , John R. Gilbert , Stefan I. Larimore , Esmond G. Ng, "A column approximate minimum degree ordering algorithm," *ACM Transactions on Mathematical Software (TOMS)*, v.30 n.3, p.353-376, September 2004.

[57] R. D. Zimmerman, D.-Q (David) Gan, "MATPOWER a MATLAB™ Power System Simulation Package," Version 3.2, Sep 21, 2007.

[58] R. Seydel, From Equilibrium to Chaos: Practical Bifurcation and Stability Analysis, New York: Elsevier, 1988.

[59] M. Huneault, A. Fahmideh-Vojdani, M. Juman, R. Calderon, and F.G. Galiana, "The Continuation Method in Power System Optimization: Applications to Economy Security Functions," *IEEE Trans. On PAS*, vol. 104, no. 1, 1985, pp. 114-124.

[60] M. Huneault and F.G. Galiana, "An Investigation of the Solution to the Optimal Power Flow Problem Incorporating Continuation Methods," *IEEE Trans. On Power System,* vol. 5, No. 1, pp. 103-110, pp. 416-423, Feb 1990.

[61] K. Iba, H. Suzuki, M. Egawa, T. Watanabe, "Calculation of the Critical Loading Condition with Nose Curse Using Homotopy Continuation Method," *IEEE Transactions on Power Systems*, Vol. 6, No. 2, pp. 584-593, May 1991.

[62] H. D. Chiang, W. Ma, R. J Thomas, and J. S. Thorp, "A Tool for Analyzing Voltage Collapse in Electric Power Systems," *Proceedings of the 10th Power Systems Computation Conference*, Graz, Austria, August, 1990.

[63] K. Iba, H. Suzuki, M. Egawa, T. Watanabe, "Calculation of the Critical Loading Condition with Nose Curve Using Homotopy Continuation Method," IEEE Trans. On Power Systems, Vol. 6, No. 2, May 1991, pp. 584-593.

[64] C. A. Canizares and F. L. Alvarado, "Point of Collapse and Continuation Methods for Large AC/DC Systems," IEEE Trans. On Power Systems, Vol. 8, No. 1, Feb 1993, pp. 1-8.

[65] H. D. Chiang, W. Ma, R. J. Thomas, and J. S. Thorp, "A Tool for Analyzing Voltage Collapse in Electric Power Systems," *Proceedings of the 10th Power Systems Computation Conference*, Graz, Austria, August, 1990.

[66] EPRI Final Report of Interactive Power Flow (IPFLOW), April 1992.

[67] J. J. Grainger and W. D. Stevenson, "Power System Analysis," New York: McGraw-Hill, 1994.

[68] J. D. Glover and M. Sarma, "Power System Analysis and Design," Boston, 1994.

[69] K. Chen, A. Hussein, and H. B. Wan, "An analysis of Seydel's test fuction methods for nonlinear power flow equations," *Int. J. Comput. Math.*, vol. 78, no. 1/2, pp. 451-470, 2001.

[70] C. T. Kelly, "Iterative Methods for linear and Nonlinear Equations," *USA: SIAM publications*, 1995.

[71] W. F. Tinney and C. E. Hart, "Power flow solution by Newton's method," *IEEE Trans. Power Apparat. Syst.*, vol 86, no 11, Nov. 1967.

[72] V. Ajjarapu and N. Jain, "Optimal continuation power flow," *Electric Power System Research35*, pp. 17-24, March 1995.

[73] A. J. Flueck, H. D. Chiang and K. S. Shah, "Investigating the installed real power transfer capability of a large scale power system under a proposed multiarea interchange schedule using CPFLOW," *IEEE Trans. On Power Systems*, vol. 11, no. 2, pp. 883-889, May 1996.

[74] H. Li, H. D. Chiang, H. Yoshida, Y. Fukuyama and Y. Nakanishi, "The generation of ZIP-V curves for tracing power system steady state stationary behaviour due to load and generation variations," *IEEE Proc. Of PES Summer Meeting*, vol. 2, pp. 647-651, Jul 1999.

[75] J. R. Shewchuk, "An introduction to the conjugate gradient method without the agonizing pain," School of Computer Science, Carnegie Mellon University, Pittsburgh, Aug 1994.

[76] W. F. Tinney and J. W. Walker, "Direct solutions of sparse network equations by optimally ordered triangular factorization," *Proceedings of the IEEE*, vol. 55, Nov 1967, pp. 1801-1809.

[77] M. A. Pai, "A new preconditioning technique for the GMRES algorithm in power flow and P-V curve calculations," *Int. J. Elect. Power Energy Syst.*, vol. 25, pp. 239-245, 2003.

[78] C. PAIGE AND M. SAUNDERS, "Solution of sparse indefinite systems of linear equations, " *SIAM J. Numer. Anal.*, Dec 1975, pp. 617-629.

[79] W. ARNOLDI, "The principle of minimized iterations in the solution of the matrix eigenvalue problem," *Quart. Appl. Math.*, Sep 1951, pp. 17-29.

[80] height 2pt depth -1.6pt width 23pt, "Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems," *SIAM J. Sci. Statist. Comput*, 13 (1992), pp. 631-644.

[81] F. Alvarado, D. Hasan, S. Harmohan, "Application of conjugate gradient method to power system least squares problems," *SIAM conference on Linear Algebra, Snowbird*, Colorado, Jun 1994.

[82] F. Alvarado, H. Dag and M. ten Bruggencate, "Block-Bordered Diagonalization and Parallel Iterative Solvers," *Colorado Conference on Iterative Methods*, Breckenridge, Colorado, April 5-9, 1994.

[83] G. H. Golub and C. F. Van Loan, "Matrix Computations," *The Johns Hopkins University Press*, USA, 1983.

[84] H. Dag and F. Alvarado, "The effect of ordering on the reconditioned conjugate gradient method for power systems applications," *Proc. Of the North American Power Symposium,* Manhattan, Kansas, pp. 202-209, Sep 1994.

[85] A. Jennings, "Influence of the Eigenvalue Spectrum on the Convergence Rate of the Conjugate Gradient Method," *J. of the Institute of Mathematics and Applications*, 20: 61-72, 1977.

[86] H. Javidi, S. McFee and F. D. Galiana, "Investigation of Eigenvalue Clustering by Modified Incomplete Cholesky Decomposition in Power Network Matrices," *Proc. of the Power System Computation Conference*, Aug. 1993.

[87] I. S. Duff and G. A. Meurant, "The Effect of Ordering on Preconditioned Conjugate Gradients," *BIT*, Vol. 29, 1989, pp. 635-657.

[88] W. F. Tinney and C. E. Hart, "Power Flow Solution by Newton's Method", *IEEE Transactions on Power Apparatus and Systems*, Vol. PAS-86, No. 11, Nov. 1967, pp. 1449-1460.

[89] B. Stott and O. Alsac, "Fast decoupled load flow", *IEEE Transactions on Power Apparatus and Systems*, Vol. PAS-93, June 1974, pp. 859-869

[90] R. van Amerongen, "A General-Purpose Version of the Fast Decoupled Load flow", *IEEE Transactions on Power Systems*, Vol. 4, No. 2, May 1989, pp. 760-770.

[91] H. Mori, H. Tanaka, "A preconditioned fast decoupled power flow method for contingency screening," *IEEE Power Industry Computer Applications Conference*, Salt Lake City, May 7-12, 1995, pp. 262-270

[92] Nigel Trevor Hawkins, " On-Line Reactive Power Management in Electric Power Systems," 1996.

[93] H. Z. Liu, Y. Li and X. Chen, "Calculation of Transmission Capability Using Continuation Power Flow", Journal of Electric Power Automation Equipment, (In Chinese) Vol. 23, No. 12, pp. 5-8, Dec. 2003.

[94] U.S.-Canada Power System Outage Task Force, "Causes and Recommendations", *Final Report on the August 14, 2003 Blackout in the United States and Canada*. Apr 2004.