Power System State Estimation and Contingency Constrained Optimal Power Flow - A Numerically Robust Implementation

by

Slobodan Pajić

A Dissertation
Submitted to the Faculty
of the
WORCESTER POLYTECHNIC INSTITUTE
in partial fulfillment of the requirements for the
Degree of Doctor of Philosophy
in
Electrical and Computer Engineering
by

April 2007

APPROVED:

Dr. Kevin A. Clements, Advisor

Dr. Paul W. Davis

Dr. Marija Ilić

Dr. Homer F. Walker

Dr. Alexander E. Emanuel

Abstract

The research conducted in this dissertation is divided into two main parts. The first part provides further improvements in power system state estimation and the second part implements Contingency Constrained Optimal Power Flow (CCOPF) in a stochastic multiple contingency framework.

As a real-time application in modern power systems, the existing Newton-QR state estimation algorithms are too slow and too fragile numerically. This dissertation presents a new and more robust method that is based on trust region techniques. A faster method was found among the class of Krylov subspace iterative methods, a robust implementation of the conjugate gradient method, called the LSQR method.

Both algorithms have been tested against the widely used Newton-QR state estimator on the standard IEEE test networks. The trust region method-based state estimator was found to be very reliable under severe conditions (bad data, topological and parameter errors). This enhanced reliability justifies the additional time and computational effort required for its execution. The numerical simulations indicate that the iterative Newton-LSQR method is competitive in robustness with classical direct Newton-QR. The gain in computational efficiency has not come at the cost of solution reliability.

The second part of the dissertation combines Sequential Quadratic Programming (SQP)-based CCOPF with Monte Carlo importance sampling to estimate the operating cost of multiple contingencies. We also developed an LP-based formulation for the CCOPF that can efficiently calculate Locational Marginal Prices (LMPs) under multiple contingencies. Based on Monte Carlo importance sampling idea, the proposed algorithm can stochastically assess the impact of multiple contingencies on LMP-congestion prices.

Acknowledgements

I would like to express my deepest appreciation and gratitude to my advisor, Dr. Kevin A. Clements. His guidance and support were essential for the development of this dissertation. I could not have imagined a better mentor than Dr. Clements. His insightful experience and editorial assistance have always been immensely helpful.

I gratefully acknowledge Dr. Paul W. Davis for his invaluable comments while patiently going over drafts and drafts of my dissertation. Without Dr. Davis's revisions, clarity of the presented research would have not been the same.

Additionally, I would like to thank Dr. Homer F. Walker for teaching me the art of numerical analysis. I am also obliged to Dr. Walker for the numerous discussions and guidance over the course of this dissertation.

I am deeply indebted to Dr. Alexander E. Emanuel for his tremendous assistance on many levels. Our rich collaboration was not only scientifically rewarding, but also inspirational on a personal level.

I'm ever thankful to Dr. Marija Ilić for encouraging me to pursue my graduate studies in electrical engineering. Without her, this scientific journey may not have occurred.

From the bottom of my heart, I wish to thank my mother Ljiljana Čolak and my sister Jelena Pajić, for their endless love, support and understanding.

Financial support for a part of this research was provided by the National Science Foundation under grant ECS-0086706.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Challenges in Power Systems Computation Applications

Large-scale electric power systems are extremely complex, and have been designed and operated conservatively through the years. At the present time, many power systems throughout the world are undergoing fundamental operational changes. Under open-access regulations, transmission owners are required to open their systems to use by other entities, including many non-utility players. What was once intended as a bridge between generation and the distribution system, transmission system became an electricity market trading floor. Many players in the game are now more oriented towards commercial goals rather than the technical. With that respect, the power grid faces many challenges that it was not designed and engineered to handle. Among challenges that modern power system computer applications have to solve are:

- new and unanticipated conditions
- atypical power flow (quick changes due to unusual modes of energy trades, such as wheeling)
- congestion
- multiple contingencies (requiring redefined reliability criteria)
- out-of-date modeling and parameter data

Computation is now heavily used in all aspects of power networks. The new restructured environment places more engineering and financial demands to operate reliably, robustly and efficiently. The market participants would also like to have dynamic information about the physical system

state, past, present, and forecast. A key challenge is to have a real-time model so that power network computations are performed on a model that resembles the current situation. When we say a real-time model we mean a "snapshot" of the system that contains redundant measurements of quantities of interest, the correct topology from which measurements are derived and accurate parameters of the elements in the model. An Energy Management System (EMS) provides a variety of measured data and computer applications for monitoring and control of the power network. When we refer to computer applications we mean the following two:

- State estimator (an on-line application)
- Contingency constrained OPF (an off-line application)

Started as engineering tool, the power system state estimator became the key data processing tool in modern EMS systems, and evolved in today's industry as a very important application for Locational Marginal Pricing algorithms for charging congestion in power networks.

Monitoring and control of power system assets is conducted through the supervisory control and data acquisition (SCADA) system. In the early days, it was believed that the real-time data base provided by SCADA could provide an operator with an accurate system view. Very soon, the deficiencies of SCADA were realized. To mention a few: hard to assure availability of all measurements at all times, measurements prone to errors, etc. A more powerful tool was needed to process collected measurements and to filter bad ones. A central master station, located at the control center, gathers information through the SCADA system. The SCADA system collects measurement data in real time from remote terminal units (RTUs) installed in substations across the power system. Typical RTU measurements include power flows (both active and reactive), power injections, voltage magnitude, phase angles and current magnitude.

While there is not much to be said that is not already known about active and reactive power and voltage magnitude measurements, voltage angle measurements are relatively new in practice. Direct measurement of voltage phase angle was impossible for a long time. In order to be valid, those measurements should be synchronized, i.e. a time reference should be provided. The global positioning system (GPS) signal made synchronization possible with accuracy better than 1 μs. A phasor measurement unit (PMU) equipped with a GPS receiver allows for synchronization of measurements, yielding accurately measured and time-stamped voltage phase angles. A study of impact of PMU measurements on state estimation and optimal placement of PMUs is given in [72]. The general conclusion is that PMUs have greatly improved observability and accuracy of voltage

angle estimates. Despite some opinions to the contrary, PMUs will not make state estimation obsolete even if they are available at every bus in the system. As we know, measurements are not perfect; thus a redundant set of measurements will still be needed in order to identify bad data.

All of these measurements can be considered dynamic since snapshots are performed every few seconds. The status of the assets (line status, breaker status etc.) as well as network parameters can be considered as static measurements. The network topology processor in Fig. 1.1 determines the topology of the network from the telemetered status of circuit breakers. Having an observable set of measurements is a necessary, although not sufficient condition, for EMS computer applications.

While it is desired, coordination across the network quite often does not happen in real-time. The reasons for not heaving real time-model are varied. While many control and monitoring functions are computer based, there are still functions handled by telephone calls between the system operator and utility control centers. It is a well known fact that control room technology is behind today's state-of-the-art in the IT world.

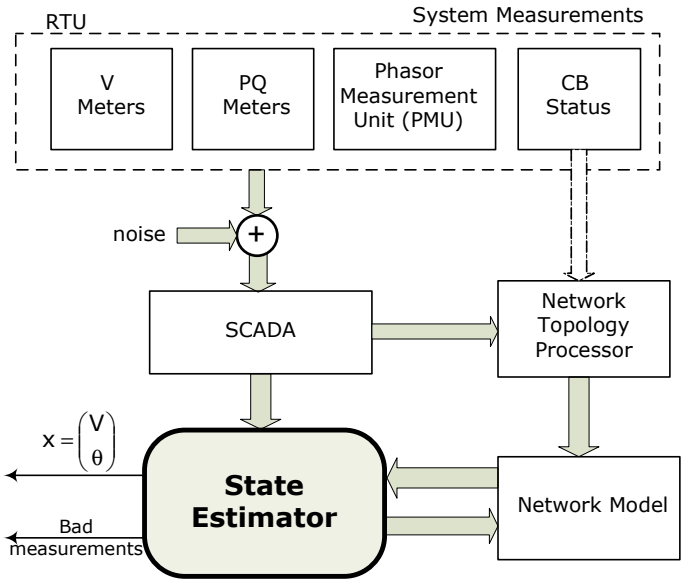

Figure 1.1: State Estimation block diagram

In particularly, equipment status from plant level to substation level is usually managed manually. Many current power systems are not capable of acquiring change of status automatically the way that, for instance, a computer operating system does. Unfortunately, it is hard to have an accurate network model in real-time. Simulations are performed frequently, whether the network model is correct or not. That means that many times simulations are performed on a network model that

does not reflect the correct network topology. While it would be nice to have a power system with the ability to auto-detect equipment status the way that computers detect plugging/unplugging external devices, it is not likely to happen soon. Situations with topology errors are common and we have to find algorithms which will successfully cope with them. That is, algorithms with the ability to detect topology errors.

One of the key EMS applications is power system state estimation. The block diagram showing the components of a modern state estimator is shown on Fig. 1.1.

To maintain a valid computer model it is essential to coordinate the computer model with the situation in the field at all times. There are situations when this objective is hard to fulfill, especially during emergencies. In those cases it is crucial to have the capability to overcome those difficulties reliably. Our work will focus on how to meet these challenges.

These improvements in power system monitoring and control are motivated by

- economics of the new market
- blackout prevention
- reliability improvement

1.1.1 Blackout Lessons

Power grids around the world have experienced a number of severe blackouts in the recent past. One is the August 2003 blackout that originated in the Midwest and affected much of the Northeastern and Midwestern United States and southern Canada. Each major blackout gives the electric power industry added attention and proves how fragile the interconnected power system really is. As in the case of the 1965 Northeast blackout, a team of national experts from the U.S. and Canada was brought together to study reasons for the blackout.

In the view of the U.S.-Canada Power System Outage Task Force, who investigated causes of the August 2003 Northeast blackout, the list of actors to blame is not that short. The impression is that the Task Force report [91] opened a Pandora's box of electric utility problems. The main factor that contributed to the blackout was the lack of tree trimming by the utility as reported by the Task Force. The well known scenario of a hot summer day, overloaded overhead lines that sagged more than usual, and ended up in vegetation that was not well maintained. Rolling outages propagated through the system and caused the blackout. To make the situation even worse, the power system monitoring tools did not work properly. The operator was unable to capture the escalating crisis at

an early stage so that affected part of the system could have been properly isolated. One of the key power system monitoring tools is the state estimator. The Midwest Independent System Operator's (MISO) state estimator at that time was not working.

The blackout did not occur instantaneously. Successive line trippings spanned an hour of agony. The critical role of computer applications in making decisions and control under blackout conditions was emphasized by Ilić *et al.* in [44].

Had the operator had a reliable and fast state estimator it is likely that widespread outage could have been avoided. Only robust state estimators that converge accurately and rapidly could be useful in these extreme situations, so that critical parts of the network could be detected and proper remedial actions taken (like shedding load) in order to prevent rolling outages. It is to be expected that such a scenario could appear more frequently in the situations when the power grid is operated near its limit.

The point of our research is not to give an optimal recommendation regarding tree trimming but to try to explore the ways of improving reliability of monitoring tools, particularly state estimator software.

The state estimator (SE) computes the static state of the system (voltage magnitude and phase angle) by monitoring available measurements. The SE has to be modeled in such a way so as to ensure that the system is monitored reliably not only in day-to-day operations, but also under the most likely conditions of system stress. The question is how to improve SE and make it more reliable, so that is more likely to capture situations like the August 14, 2003, blackout and identify critical nodes in the network.

A more robust state estimator is an essential need in the years to come. Successful SE solution relies heavily on the numerical technique used to perform the estimation. Current numerical algorithms too frequently fail to provide a successful solution. The first part in our research was to apply globalization techniques that are more reliable but cost more computationally. A subsequent part was to explore ways of reducing the computational cost of such robust SE algorithms by employing efficient modern iterative methods.

1.1.2 Reliability criteria

Electric utilities in today's market are facing many challenges and sometimes conflicting requirements. The task of maintaining reliability has been greatly complicated by the introduction of wholesale electricity markets. All players now depend on the reliability of the power grid, and

all are at risk if the grid is not reliably operated. On the one hand, the planning and operation reliability criterion is still "$N-1$" (the system must be able to withstand any single contingency event) and on the other economic forces put pressure for providing higher standards of reliability.

Security constrained optimization applications at the current stage ensure that voltage magnitude and other state and control variables are under their operating limits after the first contingency.

It has been found that traditional "$N-1$" reliability criteria for transmission and operation planning is inadequate in new (deregulated) competitive energy markets. Not just engineering (planning and operation) reliability criteria should be revisited in order to go beyond "$N-1$" but also the economic implications of such criteria must be assessed accordingly. The question is open as to who is going to pay for the higher reliability standards. Reformulating reliability policies and criteria that meet engineering, economic and regulatory needs is not an easy task.

Innovative strategies at a reasonable computational cost are required to cope with challenges that new markets impose. Reliability of the power system can be assessed either on a deterministic or a probabilistic basis. It is clear that a deterministic approach to the assessment of multiple contingencies is computationally expensive. Although it is impossible to improve reliability without additional investment, in our case computational investment, we will try to keep that investment reasonable.

After the first outage, subsequent outages are more likely to occur. Screening and ranking multiple contingencies very easily becomes a complicated task. A computational tool capable of multiple contingency modeling has two names: contingency constrained optimal power flow (CCOPF) or security constrained optimal power flow (SCOPF).

Today's market faces many new challenges. New analytical methods and algorithms should be capable of assessment of:

- multiple contingencies
- cost merits of applying more rigorous reliability criteria
- value to the customer for providing that service
- need for more rigorous security/reliability assessment

1.2 Historical Notes and Background

1.2.1 Power System State Estimation

Numerical formulation

In this section we review the current state estimation formulation and solution methods and provide motivation for further improvement. Several excellent review papers [11], [100] cover this topic in detail. When we say power system state estimation we mean the original and most widely used problem definition in practice. That is, an over determined system of nonlinear equations solved as an unconstrained weighted least-squares (WLS) problem. The WLS estimator minimizes the weighted sum of the squares of the residuals.

$$\min_{x \in \mathbb{R}^n} J(x) = \frac{1}{2} \left(z - h(x)\right)^T R^{-1} \left(z - h(x)\right)$$

where: x is the state vector; z is the measurement vector and $h(x)$ is the nonlinear vector function relating measurements to states and R is a diagonal matrix whose elements are the variances of the measurement error.

The first order necessary conditions for a minimum are that

$$\frac{\partial J(x)}{\partial x} = -H(x)^T R^{-1} \left[z - h(x)\right] = 0$$

where $H(x)$ is the measurement Jacobian matrix of dimension $(m \times n)$

$$H(x) = \frac{\partial h(x)}{\partial x}$$

Once the nonlinear measurement function $h(x)$ is linearized

$$h(x + \Delta x) \approx h(x) + H(x)\Delta x$$

the following iterative process is obtained[1]

$$\begin{aligned} \left(H^T R^{-1} H\right) \Delta x &= H^T R^{-1} \left[z - h(x)\right] \\ x^{k+1} &= x^k + \Delta x \end{aligned} \tag{1.1}$$

The symmetric matrix $H^T R^{-1} H \in \mathbb{R}^{n \times n}$ is called the *gain* or *information matrix*. Equations (1.1) are the so-called normal equations of the least-squares method and the iteration step Δx can be found only when the gain matrix is nonsingular.

[1] For simplicity, we will write $H(x)$ as H whenever clear from context

Fred Schweppe introduced WLS power system state estimation in 1969 in his classic papers [77], [76], [74]. Since then power system state estimation has been a very active research area. Besides the WLS algorithm, other state estimation methods such as decoupled WLS and Least Absolute Value (LAV) estimation were developed, but WLS is dominant in practical implementations. The overall state estimation process consists of the following steps:

1. data acquisition;
2. network topology processing;
3. observability analysis;
4. estimation of the state vector;
5. detection/identification of bad data.

An extensive bibliography of the first two decades (1968-1989) of power system state estimation was prepared by Coutto, Silva and Falcão [21]. Comprehensive treatment of modern power system state estimation can be found in books first by by Monticelli [57] in 1999 and then by Abur and Gómez Expósito in 2004 [1]. Beginning with the role of the state estimator in a security framework as one of the key modern Energy Management System (EMS) applications, they covers all parts of the state estimation process starting with power flow, problem formulation, basic solution techniques, observability, detection and identification of bad data, and robust state estimation procedures. An overview paper by Bose and Clements [11] covers the overall role of the SE in the power system control centers starting from topology processing, then goes through an overview of state estimation numerical algorithms, network observability, and bad data detection.

The subject of state estimation is vast, and we have chosen to review only those topics that are directly relevant to the rest of our dissertation. It will be hard to cover almost 40 years of active research in theory and practice of power system state estimation, and the list of contributors is long. There are many aspects of the overall state estimation process, but since the focus of this work is numerical methods for the solution of power system state estimation, at this point we will present an overview and discuss specifics as they are needed in the dissertation. Each chapter will have a background and bibliography review for the related topic.

The first approach to solving state estimation problems was the normal equation approach. More precisely, Cholesky decomposition was proposed to factor the gain matrix G ($G = H^T R^{-1} H$) in the normal equation. Then the solution is obtained by forward/backward substitution. The difficulty

with this approach was that gain matrix may be ill-conditioned, in which case the solution may fail to converge which was a major reason that other methods were sought.

The condition number (which represents the degree of system ill-conditioning) of the gain matrix in the normal equation is equal to the square of the condition number of the Jacobian (H). When H is not well conditioned, G is very ill-conditioned. Therefore, in general, squaring the Jacobian is not a good idea. The main reasons for the deteriorated condition number of the normal equation that have been cited in the literature [1] are:

- very accurate measurements (*virtual measurements*);
- large number of injection measurements;
- connection of very long transmission line (large impedance) with very short transmission line (short impedance).

Virtual measurements are measurements that do not require metering. One example is a zero injection at a switching station. Since they represent "perfect" measurements, they are characterized with very small weighting factor. In the normal equation approach, huge discrepancies between the weights renders the problem ill-conditioned. The impact of a large number of injection measurements on numerical conditioning was first observed by Gu *et al.* in [36]. Also a recent paper by Ebrahimian and Baldick [28] covers condition number analysis.

The next stage in the research was to try methods that prevent computing the gain matrix. A solution based on orthogonal transformation was first proposed by Simões-Costa and Quintana. Their first idea was based on column-wise Householder transformation [81] and the second on row-wise Givens rotations [80]. Orthogonal factorization, also known as QR factorization, of an $m \times n$ matrix H is given by

$$H = QR$$

where $R \in \mathbb{R}^{m \times n}$ is an upper trapezoidal matrix and $Q \in \mathbb{R}^{m \times m}$ is *orthogonal*. Orthogonal matrices satisfies $Q^T Q = QQ^T = I$. Discussion of the orthogonal factorization method is left for Chapter 2 where this method will be treated in detail. While one problem of ill-conditioning was solved with orthogonal transformation, other problem of fill-ins appeared. The phenomenon of turning a zero element of a sparse matrix into a nonzero element during a factorization is called *fill-in*. Originally extensive fill-ins in the process of orthogonal transformation prevent the method from being widely used. The problem of fill-ins remains to be solved.

As mentioned above Gu *et al.* studied sources of ill-conditioning and offered an alternative - the method of Peters and Wilkinson [36]. The Peters and Wilkinson method factors H and thus avoids forming $H^T R^{-1} H$. This factorization has the form

$$P_1 H P_2 = LDU$$

where: P_1 and P_2 are permutation matrices used for enhancing numerical stability and preserving sparsity, L is an $m \times n$ lower unit trapezoidal matrix, D is diagonal matrix and U is $n \times n$ upper triangular matrix. The transformed normal equation is:

$$L^T R^{-1} LUs = L^T R^{-1} \left(z - h(x)\right)$$

The above equation is solved in two stages. In the first stage Cholesky factorization of $L^T R^{-1} L$ is used resulting in

$$L^T R^{-1} L = \bar{L}\bar{D}\bar{U}$$

where $\bar{L}$ is a $n \times n$ unit lower triangular matrix and $\bar{D}$ is $n \times n$ diagonal matrix. In the second stage, above system is solved in terms of auxiliary variable y from $\bar{L}^T \bar{L} y = \bar{L}^T r$, then s is computed from $\bar{U} s = y$ via backward substitution. Although computationally more expensive than the normal equation method, the method of Peters and Wilkinson is a tradeoff between speed and stability. Improvement in conditioning of $L^T L$ compared with $H^T H$ in the normal-equation approach has been shown.

So far state the estimation problem was formulated as an unconstrained minimization problem. Extending it to a constrained optimization problem started with the work of Aschmoneit *et al.* [8]. There are buses in the network that have neither load nor generation. They are zero injection power buses. Also these measurements are so-called virtual measurements, as mentioned earlier. The idea is to use this very accurate information in order to enhance the accuracy of the estimates. Aschmoneit treated those measurement separately from the telemetered measurements and imposed them as additional constraints to the WLS problem

$$\begin{aligned} \min \quad & J(x) = \frac{1}{2} \left(z - h(x)\right)^T R^{-1} \left(z - h(x)\right) \\ \text{subject to:} \quad & c(x) = 0 \end{aligned}$$

The constrained minimization problem was then solved by the method of Lagrange multipliers. The Lagrangian ($\mathcal{L}$) is formed as:

$$\mathcal{L}(r, x, \lambda) = \frac{1}{2} \left(z - h(x)\right)^T R^{-1} \left(z - h(x)\right) + \lambda^T c(x)$$

where λ is the vector of Lagrangian multipliers. The first order necessary conditions for the optimum states that derivatives of the Lagrangian with respect to x and λ must vanish

$$\begin{aligned} \frac{\partial \mathcal{L}(x,\lambda)}{\partial x} &= -H^T R^{-1}\left[z - h(x)\right] + C^T\lambda = 0 \\ \frac{\partial \mathcal{L}(x,\lambda)}{\partial \lambda} &= c(x) = 0 \end{aligned}$$

By applying Newton's method to the above system of nonlinear equations, the following set of linear equations is solved iteratively

$$\begin{pmatrix} H^T(x^k)R^{-1}H(x^k) & C^T(x^k) \\ C(x^k) & 0 \end{pmatrix} \begin{pmatrix} s^{k+1} \\ \lambda^{k+1} \end{pmatrix} = \begin{pmatrix} H^T(x^k)R^{-1}r(x^k) \\ -c(x^k) \end{pmatrix}$$

where $C(x)$ is the constraint equation Jacobian matrix $C(x) = \partial c(x)/\partial x$ and $r(x) = z - h(x)$. The coefficient matrix above is indefinite; therefore row ordering must be employed in order to preserve numerical stability.

A similar constrained weighted least-squares problem formulation was presented by Gjelsvik, Aam and Holten in [32]. Regular measurements are imposed as constraints in the formulation where the explicit optimization variables are the measurements residuals. The method is known as the sparse tableau method or Hachtel's method:

$$\begin{aligned} \min \qquad & J(x) = \frac{1}{2} r^T R^{-1} r \\ \text{subject to:} \qquad & r = z - h(x) \end{aligned}$$

The Lagrangian function for this problem can be written as:

$$\mathcal{L}(r,x,\lambda) = \frac{1}{2} r^T R^{-1} r - \lambda^T \left(r - z + h(x)\right)$$

The necessary conditions for a minimum are given by:

$$\begin{aligned} \frac{\partial \mathcal{L}(r,x,\lambda)}{\partial r} &= R^{-1} r - \lambda = 0 \\ \frac{\partial \mathcal{L}(r,x,\lambda)}{\partial x} &= H^T \lambda = 0 \\ \frac{\partial \mathcal{L}(r,x,\lambda)}{\partial \lambda} &= z - h(x) - r = 0 \end{aligned}$$

After elimination of r and application of Newton's method, we obtain the iterative linear system

$$\begin{pmatrix} R & H(x^k) \\ H^T(x^k) & 0 \end{pmatrix} \begin{pmatrix} \lambda^{k+1} \\ s^{k+1} \end{pmatrix} = \begin{pmatrix} r(x^k) \\ 0 \end{pmatrix}$$

In this formulation ordering is required, since the coefficient matrix is again indefinite. Gjelsvik *et al.* presented numerically stable results obtained using the sparse tableau method.

Holten *et al.* compared performance of different methods (normal equations, orthogonal transformation, normal equations with constraints and Hachtels' method) for power system state estimation [40]. It has been found that orthogonal transformation (QR decomposition) is the most stable method although it has the highest computational requirements. Also it has been reported that Hachtel's method is comparable in numerical stability with orthogonal transformations.

Although numerically stable, Givens rotations can produce excessive fill-ins and therefore additional computational burden. Vempati, Slutsker and Tinney in [93] improved efficiency by employing ordering to preserve sparsity and minimize the number of intermediate fill-ins. Although there are three different ordering schemes, the most widely used is the Tinney 2 ordering scheme which employs column ordering and then uses row ordering according to the minimum column index of the row. In this form, Givens rotation establish itself as the method of choice; and it began to be used widely.

Another way of treating a virtual measurement is as a very accurate measurement with a corresponding very small variance. In other words zero injections have been modeled as measurements rather than constraints. This approach applied to the normal equation method created an ill-conditioning problem, and did not always work well in practice. Since the QR method is a numerically reliable method, it did not have any problems handling equality constraints as accurate measurements.

The power system community gained interest in interior point methods (IPM) for the solution of constrained optimization problems in early 90's. The first to apply IPM to SE problems were Clements, Davis and Frey. They explicitly included inequality constraints and solved with the IPM, first Weighted Least Absolute Value (WLAV) estimation in [17], while modeling inequality constraints in WLS SE and solving the problem with IPM started with paper [18]. They recognized that generator Var limits and transformer turns ratio constraints may be violated once state estimates were found. In order to prevent such violations, inequality constraints were added as in the

problem formulation:

$$\begin{aligned} \min \quad & J(x) = \frac{1}{2} r^T R^{-1} r \\ \text{subject to:} \quad & f(x) + s = 0 \\ & g(x) = 0 \\ & r - z + h(x) = 0 \\ & s \geq 0 \end{aligned}$$

In the IPM, the inequality constraint on the slack variable s are treated by appending a logarithmic barrier function to the Lagrangian function

$$\mathcal{L}_\mu = \frac{1}{2} r^T R^{-1} r - \mu \sum_{k=1}^{p} \ln s_k - \lambda^T \left(f(x) + s \right) - \rho^T g(x) - \pi^T \left(r - z + h(x) \right)$$

The next step is to form the Karush-Kuhn-Tucker (KKT) first order necessary conditions. The nonlinear system of KKT conditions can be solved iteratively using Newton's method. The interior point method produce iterates that are interior to the feasible region, by forcing the barrier parameter $\mu > 0$ to decrease towards zero as iterates progress. The computational experiences with the IPM method were reported and were found encouraging.

An approach to generalized state estimation that enhances robustness has been proposed by Alsaç, Vempati, Stott and Monticelli in [6]. The idea behind this formulation is to expand conventional state estimation to include topology status and network parameters as state variables. Then integrated estimation of states, status and parameters is performed. In order to be able to perform generalized state estimation a model that requires explicit representation of switching devices is needed. The authors report that generalized estimation is a more robust approach to process topology errors. A larger state vector imposes a higher computational burden on the estimator. Since parameter and status estimation are not needed at every run of an estimator, the authors suggest that its "generalized function" should be invoked only as needed.

State Estimation in practice

While it is important to follow the state-of-the-art in numerical analysis and to continually improve state estimator algorithms, it is equally important to follow how SE is implemented in practice, and what kind of infrastructural problems it is facing. A state estimator can generate an extensive amount information of the system state that is well beyond what a SCADA system is able

to do. That is a major motivation that should drive electric utility industry towards SE successful practical implementation.

The whole process of state estimation is a very large and complex hardware-software system and today is usually based in an Independent System Operator (ISO) control center. Real-time implementation and practical experience have been reported in a few papers describing how SE performs in practice on day-to-day operations. Dy Liacco in [27] stressed experiences with state estimators in EMS control centers and covers limitations like critical measurements, topology errors etc. The panel discussion at the 2005 IEEE PES General Meeting addressed some of the challenges faced by the SE in practice and stressed why SE still did not achieve its expected role in the electric utility industry. Among these papers was [2] by Allemong, who emphasized the importance of three basic categories needed for successful implementation. They are:

1. A redundant, reliable and accurate measurement set
2. Accurate network topology, constructed from the real-time status of switching elements
3. Accurate parameters for the network elements

Practitioners agreed that some issues that hinder state estimation in operation are:

- Incorrect topology or topology errors in the model (changes in topology occur continuously)
- Incorrect model parameters
- Inadequate or faulty telemetry
- Inconsistent phase metering
- Meter placement errors (inconsistency between meter placement in the field and in the computer model)

A typical problem is the incorrect assignment of a flow measurement to a piece of equipment. Many times a flow measurement is actually the sum of flows on two (or more) pieces of equipment. It is discouraging to see that the problems SE has been facing since its early implementation still exist and even today are not resolved. None of the above issues are related to the SE algorithm itself; they are rather related to the infrastructure for state estimation. Although the above problems deserve serious attention, besides recommendations, researchers cannot do much. What researchers can do is to follow the state-of-the-art in robust numerical analysis algorithms and apply them to

the SE problem in hopes of overcoming infrastructural weaknesses. Also, economic requirements of the electricity market may make these deficiencies less tolerable.

The Role of the State Estimator in Real-Time Energy Market

The primary driver behind deregulation and transmission system open access is the facilitation of effective competition in the generation sector of the power system. Under the regulated electricity market, it was the responsibility of the integrated utility to assure stable and secure grid operation. After deregulation, the control function was separated from the utility and granted to an independent entity. The Independent System Operator (ISO) is an independent, non-profit organization that administers the deregulated electricity market and oversees the security of the electric power grid.

The larger control area of the ISO has increased the need for computer systems to control the interconnected transmission grid in order to assure its reliability and market efficiency. The nature of the new real-time market monitoring is similar to the nature of system monitoring under a vertically integrated system. It has been recognized for quite some time that currently employed numerical algorithms in even the most advanced control centers are not fully adequate to ensure reliable and efficient service. In today's deregulated energy market, the state estimator becomes an increasingly critical application. More and more power markets are moving from zonal to Locational Marginal Price (LMP) based congestion management. A critical point in that move is having a reliable state estimator as a part of the real-time market system. Not just LMP, but the accuracy of many other applications like contingency analysis and dispatch depend on high quality estimates provided by state estimator.

Doudna and Salem-Natarajan in [26] discuss issues facing the SE at the ISO/RTO organization level in California (CAISO). One of the major challenges the ISO is facing is network modeling. The ISO/RTO are in charge of monitoring the system; they do not own the transmission system. The challenge that they are facing is that they must rely on the separate transmission owners to supply the associated network models, measurements, and outage information necessary for successful operation of the real-time state estimator.

Many parts of the network lack telemetry. In particular, the lack of real-time status measurements present a problem in running the SE. An additional problem for CAISO is receiving data from various entities. Many times the measurement sign convention is not consistent from one entity to another. Doudna and Salem-Natarajan emphasize that improvement in real-time telemetry

data and sign convention standards across the industry as a whole are essential elements to achieve reliable SE solution.

1.2.2 State Estimation - our research direction

Considering the state of SE today, some issues require research and some of them just more discipline in implementing the SE in practice. As far as the state of the research is concerned, existing methods are improved and new methods are being proposed constantly. The good news for researchers is that not all numerical techniques have been explored. Even though decades have passed researchers are still seeking computationally reliable efficient state estimator.

Throughout this brief survey of existing methods and formulations one can notice a common denominator for almost all of them. Once the first-order necessary conditions are imposed upon the set of nonlinear equations, the resulting problem is solved via Newton's method. Algorithms based on Newton's method have dominated the power system state estimation community for decades. From the practical point of view, however, there are more efficient and robust methods. Those methods lie in the family of trust-region methods (TRM) and recently have become very popular in the optimization community. Development of the trust-region method has focused primarily on the solution of unconstrained optimization problems such as the state estimation problem. TRM is based on a globalization of Newton's method which is very often the key to the success (finding a global minimum) of the algorithm. The TRM has not been tested on the power system state estimation problem prior to this research.

It is widely known that Newton's method performs very well when the iterates are near the solution. So in that region there is no reason to use anything else but Newton's method. And that is exactly what the trust-region method does. When Newton's method performs well a step is chosen according to it, as soon as a successful step can not be found, the trust-region iteration is employed. The algorithm provides an automatic choice between the Newton and the trust region method.

We start Chapter 2 with a review of the state-of-the-art of the QR algorithm, and we give an example under which this method in the presence of topology error does not perform reliably.

Our contribution is in trust region methods and further improvement with modern Krylov iterative methods. Review of the the trust region literature will be left for chapter 2, and review of the Krylov subspace methods will be left for Chapter 3.

1.2.3 Optimal Power Flow (OPF) - problem formulation

The goal of the Optimal Power Flow (OPF) is to calculate a state of the power system and values of the control variables which minimize a given objective function (e.g. generation cost, network losses, etc.) and at the same time satisfy all constraints imposed on the problem. The classical OPF (also called the base-case) can be stated as the following nonlinear programming problem:

$$\begin{aligned} \min \quad & c(x,u) \\ \text{subject to:} \quad & g(x,u) = 0 \\ & f(x,u) \leq 0 \end{aligned} \tag{1.2}$$

$$x = \begin{pmatrix} v \\ \theta \end{pmatrix} \in \mathbb{R}^{2n}, \qquad u = \begin{pmatrix} p_g \\ q_g \\ t_b \\ \phi \end{pmatrix} \in \mathbb{R}^{n_u}$$

where: x is a vector of state variables (voltage magnitude v and phase angles θ), u is a vector of controllable variables (generator outputs, adjustable transformers), $g(x,u)$ is a nonlinear vector function whose elements are $g_i(x,u)$, where $i \in \mathcal{E}$, and represent power balance equations at each node in the network and $f(x,u)$ is a vector whose elements are $f_i(x,u)$, where $i \in \mathcal{I}$, are limits imposed on the system.

The most common objective functions include minimum cost of operation, minimum active power losses, minimum deviation from a specific operating point, minimum number of controls rescheduled, etc. The objective function usually depends on variables with direct cost u (power generation, load shedding, etc.) and variables without direct cost x (voltage magnitude). The objective that is most widely used is the cost of operation, which in the security-constrained framework accounts for cost of generation and load shedding. One way to model load shedding is as a "very expensive negative generation", since otherwise the cheapest solution will be to shed as much load as possible. The cost of thermal units is derived from the heat-rate curves which are sometimes far from convex. Convexity of the objective function is one of the assumptions for the optimization method employed in the solution of the OPF problem; hence cost curves are usually approximated as convex polynomials, most often quadratic:

$$c_g(p_g) = a \cdot p_g^2 + b \cdot p_g + c$$

where p_g is the MW (or per-unit) output of the generator and a, b and c are quadratic polynomial coefficients. Other approximations, such as using an arbitrary number of line segments, are used as well.

OPF incorporates a wide variety of constraints that are formulation-specific. Constraints that are important in one may not be important in another formulation. The set of constraints, as seen from the formulation (1.2), can be divided into equality and inequality constraints. The equality constraint set typically consists of power balance equations (both active and reactive) at each node of the network. In general, inequality constraints can be classified in three categories:

1. dispatchable (active and reactive power, tap changing and phase shifting transformers)
2. variables (voltage magnitude and phase angles)
3. functions of variables (line flows based on thermal limits)

Generators are rated by the maximum apparent power (S^{max}) which they can produce. The combination of P and Q produced by a generator must obey the apparent circle equation $P^2+Q^2 \leq S^{max}$. In practice, this condition is usually approximated so that each generator in the system is subject to the box constraints:

$$p_i^{min} \leq p_i \leq p_i^{max}$$
$$q_i^{min} \leq q_i \leq q_i^{max}$$

Besides generators, transformers provide an additional means of control of the flow of both active and reactive power. There are two types of controllable transformers, tap changers and phase shifters, although some transformers regulate both the magnitude and phase angle. Controllable transformers are those which provide a small adjustment of voltage magnitude, usually in the range $\pm 10\%$, or which shift the phase angle of the line voltages. A type of transformer designed for small adjustments of voltage rather than for changing voltage levels is called a regulating transformer.

1.2.4 OPF Solution Techniques

The large number of variables and limit constraints make the OPF a computationally demanding nonlinear programming problem. Since OPF has been around since the early '60s, many methods have been tried. The choice of a solution method is particularly important. It deserves careful analysis and depends on many factors (accuracy, speed, storage, etc.). And as usually happens, there is no method that fits all applications and that has all desirable properties.

The classical OPF formulations were pioneered by Carpentier [14] and Dommel and Tinney [25]. Their method was based on the use of a penalty function to account for constraints, the solution of the power flow by Newton's method, and the optimal adjustment of control variables by the gradient method.

An extensive survey of the publications in the field of optimal power flow from the early days up to the year 1991, with a classification based on methods of optimization technique used, is given in Huneault and Galiana in [41]. A comprehensive review of the OPF algorithms was prepared by Glavitsch and Bacher in [33].

There are two main approaches to the OPF problem formulation:

a) the exact nonlinear formulation or so-called full AC formulation

b) the linearized problem formulation (DC or incremental formulation)

Equality constraints are treated by the method of Lagrange multipliers. The Lagrangian function of the problem (1.2), whose inequality constraints are transformed into equality constraints by means of the *slack* variable s is:

$$\mathcal{L} = c(x,u) + \lambda^T g(x,u) + \pi^T (f(x,u) + s)$$

where λ and π are vectors of Lagrange multipliers. The first-order (necessary) conditions, or Karush-Kuhn-Tucker (KKT) conditions for the solution are:

$$\begin{aligned}
\nabla_x \mathcal{L} &= \nabla_x c(x,u) + G_x^T \lambda + F_x^T \pi = 0 \\
\nabla_u \mathcal{L} &= \nabla_u c(x,u) + G_u^T \lambda + F_u^T \pi = 0 \\
\nabla_\lambda \mathcal{L} &= g(x,u) = 0 \\
\nabla_\pi \mathcal{L} &= f(x,u) + s = 0 \\
\Pi s &= 0 \\
s, \pi &\geq 0
\end{aligned} \tag{1.3}$$

where:

$$G_x = \frac{\partial g(x,u)}{\partial x} \in \mathbb{R}^{2n \times 2n}, \qquad G_u = \frac{\partial g(x,u)}{\partial u} \in \mathbb{R}^{2n \times n_u}$$
$$F_x = \frac{\partial f(x,u)}{\partial x} \in \mathbb{R}^{n_c \times 2n}, \qquad F_u = \frac{\partial f(x,u)}{\partial u} \in \mathbb{R}^{n_c \times n_u}$$

and

$$\Pi = \mathrm{diag}(\pi)$$

The KKT equation $\Pi s = 0$ is known as the *complementary slackness* condition.

Iterative techniques are employed to solve nonlinear programming OPF problems. A sequence of subproblems, either linear or quadratic approximations to the original problem, are defined at each iteration. Methods are usually applied to an *augmented Lagrangian* that combines the requirement of optimality and feasibility in a single objective. Lagrangian is augmented by a penalty or barrier function which adds a high cost for either infeasibility or for approaching the boundary of the feasible region via its interior. The penalty and barrier term vanishes at the solution.

Sequential linear programming methods

Attractive for their speed and flexibility, linear programming methods gained much attention for application in the nonlinear world of OPF. Sequential linear programming (SLP) optimization is performed on piecewise-linear approximation of the quadratic cost function subject to an incremental linearization of the network constraints. The general form of the SLP problem is

$$\begin{aligned} \min \quad & c_x^T \Delta x + c_u^T \Delta u \\ \text{subject to:} \quad & G_x \Delta x + G_u \Delta u = -g(x,u) \\ & F_x \Delta x + F_u \Delta u \leq -f(x,u) \end{aligned}$$

where c_x and c_u are vectors of cost coefficients. An incremental linearization of the network load flow problem yields power balance equations. The sequential linear programming approach requires an outer linearization loop wherein the constraints and objective function are linearized. The linearized equations are quite sparse and have the sparsity structure of the network bus admittance matrix. By eliminating state variables from the problem using distribution factors, as proposed by Stott and Hobson in [86], results in a reduced problem formulation of the form:

$$\begin{aligned} \min \quad & c^T \Delta u \\ \text{subject to:} \quad & a^T \Delta u = b \\ & D \Delta u \leq d \end{aligned}$$

where the primary variables are controllable unit generations. This formulation has a single equality constraint and set of inequality constraints. It is similar to the economic dispatch problem, augmented with set of inequality constraints. Unfortunately, D has a large number of rows and is dense. Typically, very few of the inequality constraints are binding. This characteristic can be

exploited with the active set method that will be discussed in Chapter 5. Solution methods for the LP-based OPF are discussed by Stott and Hobson in [86] and by Stott and Marinho in [87].

Some methods use an entirely linearized system model, neglecting reactive power and voltage constraints and accepting MW-flow accuracy limitations of the DC load flow.

A method that exploits some physical properties of active and reactive power, has been proposed by Stott and Alsaç in [84] and is known as fast decoupled load-flow. To explain the idea behind this method, consider active and reactive power linearized about a given operating point:

$$\Delta P_i = \sum_{k=1}^{n} \frac{\partial P_i}{\partial \theta_k} \Delta\theta_k + \sum_{k=1}^{n} \frac{\partial P_i}{\partial V_k} \Delta V_k$$

$$\Delta Q_i = \sum_{k=1}^{n} \frac{\partial Q_i}{\partial \theta_k} \Delta\theta_k + \sum_{k=1}^{n} \frac{\partial Q_i}{\partial V_k} \Delta V_k$$

or in a matrix form

$$\begin{pmatrix} \Delta P \\ \Delta Q \end{pmatrix} = \begin{pmatrix} H & N \\ J & L \end{pmatrix} \begin{pmatrix} \Delta\theta \\ \Delta V \end{pmatrix}$$

The above equations represent an incremental model, meaning that the system is linearized about an initial system operating point, which is usually provided in real time by a state estimator or in off-line studies by an AC load flow. The fast decoupled formulation is obtained by neglecting the coupling submatrices N and J according to the following assumptions:

- insensitivity of real power to changes in voltage magnitude $\frac{\partial P}{\partial V} \ll \frac{\partial P}{\partial \theta}$
- insensitivity of reactive power to changes in phase angle $\frac{\partial Q}{\partial \theta} \ll \frac{\partial Q}{\partial V}$

The fast decoupled load-flow equations are given by:

$$\begin{aligned} \Delta P/V &= B'\Delta\theta \\ \Delta Q/V &= B''\Delta V \end{aligned}$$

where elements of matrices B' and B'' are:

$$B'_{ij} = \begin{cases} -\frac{1}{x_{ij}} & i \neq j \quad \text{assuming a branch from } i \text{ to } j \text{ (zero otherwise)} \\ \sum_{k=1}^{n} \frac{1}{x_{ik}} & i = j \end{cases}$$

$$B''_{ij} = \begin{cases} -\frac{x_{ij}}{r_{ij}^2 + x_{ij}^2} & i \neq j \quad \text{assuming a branch from } i \text{ to } j \text{ (zero otherwise)} \\ \sum_{k=1}^{n} \frac{x_{ik}}{r_{ik}^2 + x_{ik}^2} & i = j \end{cases}$$

Both matrices B' and B'' are real, sparse, and have constant elements, meaning that they need to be factored only once in the algorithm. In many practical cases, accuracy of the LP, initially proposed to improve computing speed, has proved to be adequate. Advancements being made in LP-based OPF like cost curve modeling, handling infeasibility, and loss-minimization were reported by Alsaç *et al.* in [3].

Newton's method

An extensive survey of the application of Newton's method to the power flow solution is provided by Tinney and Hart in [89]. Solution of the classical OPF formulation defined by (1.2) by Newton's method was presented by Sun *et al.* in [88]. That algorithm begins with the standard step of forming the Lagrangian function by imposing equality constraints and penalty function in terms of inequality constraints. The set of KKT conditions (1.3) in this approach is solved by Newton's method, resulting in the system that has to be solved at each iteration:

$$\begin{pmatrix} H & -J^T \\ J & 0 \end{pmatrix} \begin{pmatrix} \Delta z \\ \Delta \lambda \end{pmatrix} = \begin{pmatrix} -\partial \mathcal{L}/\partial z \\ -\partial \mathcal{L}/\partial \lambda \end{pmatrix}$$

where Δz is a vector of incremental state Δx and control Δu variables, and $\Delta \lambda$ is the vector of incremental Lagrangian multipliers. Factorization and solution of the above problem requires four times as much computational effort compared with the power flow problem. In order to save computational work per iteration, Sun *et al.* also presented a decoupled version based on [84] that requires approximately the same amount of computational effort as Newton power flow. It was reported in [3] that in the full nonlinear version, convergence difficulties were encountered when contingency constraints were included.

Sequential quadratic programming (SQP) methods

Probably the most powerful, highly regarded method for solving nonlinear optimization problems involving nonlinear constraints is sequential quadratic programming (SQP), also called successive quadratic programming. The SQP method generates a sequence of iterates, each of which is the minimizer to a quadratic subproblem that is a local model of the initial nonlinear constrained problem. For more details on the SQP method, see Bertsekas [10].

The SQP method for the solution of the OPF problem defined by (1.2) was proposed first by Burchett *et. al* in [12]. The method linearizes the KKT conditions at each iteration of the original

nonlinear problem rather than linearizing the problem itself. Since linearized KKT proceeds from the quadratic programming problem, the method is called sequential quadratic programming (SQP).

The SQP subproblems contain exact first- and second-order derivatives of the nonlinear objective function and the linearized power flow equations. Like sequential LP algorithms, SQP algorithms have an outer linearization loop and an inner optimization loop.

First linearize the KKT conditions given by (1.3)

$$\begin{aligned} W_{xx}\Delta x + W_{xu}\Delta u + G_x^T\lambda + F_x^T\pi &= -\nabla_x c(x,u) \\ W_{ux}\Delta x + W_{uu}\Delta u + G_u^T\lambda + F_u^T\pi &= -\nabla_u c(x,u) \\ G_x\Delta x + G_u\Delta u &= -g(x,u) \\ F_x\Delta x + F_u\Delta u + s &= -f(x,u) \\ \Pi s &= 0 \end{aligned}$$

W_{xx}, W_{xu}, W_{ux} and W_{uu} represent the second order derivatives of the Lagrangian function with respect to control and state variables and are defined as follows:

$$\begin{aligned} W_{xx} &= \nabla^2_{xx} c(x,u) + \sum_{i=1}^{n} \frac{\partial^2 g_i}{\partial x^2}\lambda_i + \sum_{i=1}^{n_c} \frac{\partial^2 f_i}{\partial x^2}\pi_i \\ W_{xu} &= \nabla^2_{xu} c(x,u) + \sum_{i=1}^{n} \frac{\partial^2 g_i}{\partial x \partial u}\lambda_i + \sum_{i=1}^{n_c} \frac{\partial^2 f_i}{\partial x \partial x}\pi_i \\ W_{ux} &= W_{xu}^T \\ W_{xx} &= \nabla^2_{uu} c(x,u) + \sum_{i=1}^{n} \frac{\partial^2 g_i}{\partial u^2}\lambda_i + \sum_{i=1}^{n_c} \frac{\partial^2 f_i}{\partial u^2}\pi_i \end{aligned}$$

The corresponding Lagrangian is

$$\begin{aligned} \mathcal{L} &= \begin{pmatrix} \nabla_x c^T(x,u) & \nabla_u c^T(x,u) \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta u \end{pmatrix} + \frac{1}{2} \begin{pmatrix} \Delta x^T & \Delta u^T \end{pmatrix} \begin{pmatrix} W_{xx} & W_{xu} \\ W_{ux} & W_{uu} \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta u \end{pmatrix} \\ &+ \lambda^T \left(G_x\Delta x + G_u\Delta u + g(x,u)\right) \\ &+ \pi^T \left(F_x\Delta x + F_u\Delta u + s + f(x,u)\right) \end{aligned}$$

Now we can formulate a quadratic programming subproblem given the Lagrangian function above.

The linearized KKT conditions are the KKT conditions for the following quadratic problem (QP):

$$\begin{aligned} \min \quad & \begin{pmatrix} \nabla_x c^T(x,u) & \nabla_u c^T(x,u) \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta u \end{pmatrix} + \frac{1}{2} \begin{pmatrix} \Delta x^T & \Delta u^T \end{pmatrix} \begin{pmatrix} W_{xx} & W_{xu} \\ W_{ux} & W_{uu} \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta u \end{pmatrix} \\ \text{subject to:} \quad & G_x \Delta x + G_u \Delta u = -g(x,u) \\ & F_x \Delta x + F_u \Delta u \leq -f(x,u) \end{aligned}$$

If we define

$$\nabla c(x,u) = \begin{pmatrix} \nabla_x c(x,u) \\ \nabla_u c(x,u) \end{pmatrix} \qquad H = \begin{pmatrix} W_{xx} & W_{xu} \\ W_{ux} & W_{uu} \end{pmatrix} \qquad \Delta z = \begin{pmatrix} \Delta x \\ \Delta u \end{pmatrix}$$

and also

$$G = \begin{pmatrix} G_x & G_u \end{pmatrix} \qquad F = \begin{pmatrix} F_x & F_u \end{pmatrix}$$

then the Lagrangian function is

$$\mathcal{L} \quad = \quad \nabla c^T(z)\Delta z + \frac{1}{2}\Delta z^T H \Delta z + \lambda^T \left(G\Delta z - g(z)\right) + \pi^T \left(F\Delta z + s + f(z)\right)$$

which is the Lagrangian of the following quadratic subproblem that we have to solve at each iteration:

$$\begin{aligned} \min \quad & \nabla c^T(x,u)\Delta z + \frac{1}{2}\Delta z^T H \Delta z \\ \text{subject to:} \quad & G\Delta z = g(z) \\ & F\Delta z \leq f(z) \end{aligned}$$

Therefore, at each outer iteration the problem is approximated as a quadratic objective function with a linear constraint set approximated at the current iterate x. The quadratic objective function models the curvature of the Lagrangian. This SQP problem is solved iteratively until convergence is attained. Burchett *et al.* in [12] proposed to apply Newton's method.

Interior Point Methods (IPM)

An interior point method was developed by Nerendra Karamarkar in 1984 for linear programming, although many of the component ideas were known earlier. The algorithm used for years for solving linear programming problems has been the simplex method, which moves from one vertex of the feasible region to another while constantly attempting to improve the value of the objective

function. An interior point method implies that progress towards a solution is made through the interior of the feasible region rather than its vertices. A general reference for interior point methods is Wright [98]. The framework for developing an interior point method has three parts:

- A barrier method for optimization with inequalities
- The Lagrange method for optimization with equalities
- Newton's method for solving the KKT conditions

After the transformation of inequality into equality constraints by introducing slack variables, one augments the cost function with a barrier function. The barrier or penalty function accommodates nonnegativity constraints on slack variables. A barrier function is continuous and grows without bound as any of the slack variables approach 0 from positive values (from the interior of their feasible region). The most common example of a barrier function and the form we will use is

$$b(\mu, s) = -\mu \sum_{i=1}^{n_c} \ln s_i$$

where $\mu >$ is a scalar parameter called the *barrier parameter*. The value of μ goes to zero as the solution of the optimization algorithm progresses. After introducing the barrier function, we can write the modified OPF formulation:

$$\begin{aligned} \min \quad & c(x,u) - \mu \sum_{i=1}^{n_c} \ln s_i \\ \text{subject to:} \quad & g(x,u) = 0 \\ & f(x,u) + s = 0 \end{aligned}$$

The Lagrangian function of this problem is:

$$\mathcal{L}_\mu = c(x,u) - \mu \sum_{i=1}^{n_c} \ln s_i + \lambda^T g(x,u) + \pi^T \left(f(x) + s\right)$$

The complementary slackness condition in the primal-dual interior point method formulation is replaced by:

$$\Pi s = \mu e$$

where e is a vector of ones of appropriate dimension. Solving the SQP OPF problem by an interior point method was proposed by Nejdawi, Clements and Davis in [63] and further discussed in [62], where more details are found. An extension of that method to include the CCOPF formulation appears in Pajić [69].

Constraint relaxation method

Needless to say, if the correct binding inequalities are known and if they do not change from iteration to iteration, the OPF problem would be much easier. However, the binding inequality set is not known *a priori*. Usually, the number of inequalities imposed on the problem is large, and to model all of them will slow down the method. The term *active constraint* will be used to designate an inequality constraint that is satisfied exactly at the current point (x, u), and the set of all constraints active at a given point will be referred to as the *active set* $\mathcal{A}(x, u)$ at that point

$$\mathcal{A}(x, u) = \{i \in \mathcal{I} \mid \quad f_i(x, u) = 0\}$$

The set of constraints whose indices lie in the active set are said to be *active*, or *binding*, while the remainder are *inactive*. The challenge of any efficient algorithm for constrained minimization is to identify and model only active constraints.

Exploitation of an active set method for the OPF started with Stott in [86] relative to linear programming formulations, and was further discussed by Sun *et al.* in [88] and Burchett *et. al* in [12] in a nonlinear programming framework.

A method that only models active constraints is called a *constraint relaxation method* or an *active set method*. In this technique, we ignore constraints until they are violated. Mathematically, that means that Lagrange multipliers corresponding to inactive constraints are not considered in the problem since they are zero; only when the inequality becomes active is the corresponding multiplier is nonzero.

Each iteration begins with testing for new active constraints. Once a constraint becomes active, it is considered active for the reminder of the iterative process, thus avoiding the additional process of taking it out. Generally, only a small percentage of the total transmission constraints become active, greatly reducing the size of the OPF problem. Numerical examples presented by Kimball *et al.* in [51] show significant reduction in problem size achieved in practice by the active set method. The heuristic of adding to the active set just the most violated of the newly active constraints was proposed by Stott in [86] and has proven to be very efficient.

1.2.5 Contingency Constrained OPF

Contingencies, in power system terminology, are unpredictable disturbances to the transmission or generation facilities. It has been recognized that with the basic OPF formulation, it may not be possible to keep the system in a normal state after a contingency occurs, or even when it is

possible, the cost of such a solution may be very high. Contingency Constrained OPF (CCOPF), also called Security Constrained OPF (SCOPF) dispatch, guarantees that the system will operate successfully and optimally under the base case and the contingency case.

CCOPF is a cornerstone security application in modern power systems. A given OPF problem or so called base case, is expanded to account for credible contingencies and the problem is solved as a single entity. The mathematical formulation of the general contingency constrained OPF is as follows:

$$\begin{aligned} \min \quad & c(x,u) \\ \text{subject to:} \quad & g(x,u) = 0 \\ & f(x,u) \leq 0 \\ & g_\omega(x_\omega, u_\omega) = 0 \qquad \omega = 1,\ldots,K \\ & f_\omega(x_\omega, u_\omega) \leq 0 \qquad \omega = 1,\ldots,K \end{aligned} \tag{1.4}$$

where:

x, u	pre-contingency state and controls;
x_ω, u_ω	post-contingency state and controls;
$g(x,u)$	power balance equations for base case;
$f(x,u)$	set of inequality constraints for base case;
$g_\omega(x_\omega, u_\omega)$	power balance equations for each contingency case;
$f_\omega(x_\omega, u_\omega)$	set of inequality constraints for each contingency case;
ω	is the set of possible contingencies;

In general, $f_\omega(x_\omega, u_\omega)$ are contingency limits or security constraints that impose post-disturbance limits and may be substantially different from base case limits. The computational times for contingency constrained OPF are considerably longer than for base-case OPF.

The first paper that extended the Dommel-Tinney OPF formulation to include outage-contingency constraints into the method to give an optimal steady-state-secure system operating point is Alsaç and Stott [5]. The evolution of CCOPF algorithms follow the same path as the OPF. Linear programming formulations were presented by Stott *at al.* in [86] and [87]. Linearized CCOPF is particularly well suited for the contingency framework, since it is very easy to modify constant real matrices to account for line outages, a process that will be explained and thoroughly exploited in Chapter 4.

In a CCOPF algorithm, more often than not, more expensive generators have to be dispatched

and less expensive generators set to lower output in response to a contingency. Therefore, as in real life, an increase in security comes with an increase in cost of operation. Nonetheless, operating cost can be controlled to some extent by corrective actions. In that respect, the CCOPF can be formulated on two ways:

- so called *safe* or *preventive* contingency constrained OPF, which does not allow any rescheduling of controls in response to contingency;
- CCOPF with *corrective* rescheduling, which allows control actions shortly after the occurrence of the contingency

Corrective rescheduling is accomplished by means of fast-acting control actions taken before the slow control actions. Examples are:

- fast-acting controls: synchronous machine speed governors, synchronous machine excitation, load shedding, etc.
- slow-acting controls: transformer taps, area interexchange control, etc.

By considering the corrective action formulation, (1.4) is expanded to include so-called ramp-rate constraints or coupling constraints of the general form:

$$h(u, u_\omega) \leq 0$$

These constraints recognize that the range of adjustment of certain controls is determined by their setting at the time of the contingency. They act as a "bridge" between the base and the post-contingency case. In the algorithm they are modeled as box inequality constraints:

$$\underline{\Delta} \leq u - u_\omega \leq \bar{\Delta} \qquad \omega = 1, \ldots, K$$

where $\underline{\Delta}$ and $\bar{\Delta}$ are lower and upper ramp-rate limits. The ramp rate of generators is usually defined as a percentage of generator capacity (i.e., 10% to 15%) The idea of control actions was first presented by Stott and Hobson in [86] in the LP framework.

An excellent simple example of corrective economic dispatch is given by Monticelli *et al.* in [58]. It has been shown that corrective methods provide the same level of security as preventive methods but with the lower operating cost. In [58] the mathematical framework in which corrective CCOPF was solved based on Bender's decomposition.

It is important to understand that control actions that are essential for economic rescheduling are both active and reactive. Many times the contingency reactive constraints impose a cost penalty on MW dispatch. An example that emphasizes this important point is given in [3]. As in the OPF framework, active set methods are employed; in CCOPF, they consume a significant part of the running time of the algorithm.

Stott, Alsaç and Monticelli [85] provide a comprehensive treatment of all aspects of security analysis in the CCOPF framework.

For quite some time, the practice has been to optimize for single contingencies. Also the philosophy of CCOPF employed in practice has been preventive control rather than corrective. That is the way locational marginal prices are determined. One of the requirements of the new market is to handle a large contingency list and to identify critical contingencies in it. Screening and ranking multiple contingencies becomes a complicated task. New situations need development of new algorithms, and we will address that issue and propose a solution using importance sampling.

1.2.6 CCOPF in Today's Market

Since the OPF is a problem that combines engineering constraints and economic objectives for system operation, it has paramount importance in today's market. Many economic quantities like Locational Marginal Prices (LMP), congestion charges, and so on are derived from the OPF algorithm.

In a rapidly changing restructured power industry, market participants need to use the results of CCOPF in order to become more competitive. Extensive OPF simulations are performed by ISOs to anticipate worst-case system problems. Most abnormal voltage conditions are anticipated off-line by contingency screening algorithms, and solution of CCOPF is supposed to prepare for the worst-case contingencies.

Locational Marginal Prices are obtained directly from the solution of any OPF calculation. OPF-based algorithms are also used to access the cost of transmission congestion which emerges as the difference in energy prices between locations connected by a line whose flow has hit its limit.

One of the most difficult tasks on the road toward efficient transmission is the problem of managing and valuing uncertainties. In the past, reliability-related uncertainties have been managed in a somewhat conservative, preventive way. Those costs were distributed on a *pro rata* basis to all customers. Uncertainties are not just system related as in the past. In today's market it is hard to distinguish between system-related and market-related uncertainties. For example, is a generator

unavailable due to maintenance or because its owner does not want to participate in the marked since the price is too low [42]?

1.2.7 CCOPF - our research direction

The operation of a large interconnected system to ensure reliable operation at minimum cost is a very complex problem. The objective is designing an algorithm that will be able to handle multiple-contingencies in computationally and economically efficient manner.

Part of this dissertation presents the sequential quadratic programming technique applied to CCOPF [69], combined with the method of importance sampling in order to solve the stochastic OPF. It is widely recognized that it is impossible to model all possible contingencies. Instead, we employ Monte Carlo importance sampling techniques to obtain an estimate of the expected value of multiple-contingency operating costs. Recent blackouts warn us that there is a need for clever stochastic algorithms able to assess multiple outage scenarios having potentially catastrophic consequences. The objective in importance sampling is to concentrate the random sample points in critical regions of the state space. In our case that means that single-line outages that cause the most "trouble" will be encountered more frequently in multiple-line outage subsets.

1.3 Contents

This dissertation is organized in six chapters that are divided into two main parts. The chapters are:

1. Introduction
2. Power System State Estimation via Globally Convergent Methods
3. Newton-Krylov Methods in Power System state Estimation
4. The Use of Importance Sampling in Stochastic OPF
5. A Formulation of the DC Contingency Constrained OPF for LMP Calculations
6. Conclusion and Future Work

The first part presents further improvement in state estimation (Chapters 2 and 3), and the second part treats several implementations of CCOPF in a stochastic multiple contingency framework and LMP calculation under multiple contingencies (Chapters 4 and 5).

- In Chapter 1 we presented a general overview of power system state estimation and contingency constrained optimal power flow, a motivation for further research into more reliable computational tools, and present an historical review of the formulations and methods employed for both problems.
- In Chapter 2 the theory and implementation of the TRM method and critical implementation points are addressed and the algorithm is developed. The performance of the TRM method is tested on the standard IEEE network cases and results are discussed thoroughly.
- In Chapter 3 power system state estimation is solved by one of the most robust Krylov subspace methods for solving least-squares problems, the so-called LSQR method.
- In Chapter 4 sequential-quadratic programming (SQP) contingency constrained optimal power flow is combined with the method of Monte Carlo importance sampling in order to solve the stochastic optimal power flow.
- In Chapter 5 we develop LP-based CCOPF formulation that can efficiently handle multiple contingencies. The novel formulation can be used in importance sampling framework to produce an estimate of LMP-based congestion price of multiple contingencies.
- In Chapter 6 presents a brief summary of this research and a discussion of possible future work.
- The Appendix A provide the network test cases that are used throughout this research.
- The Appendix B covers important theorems used for the reduced problem formulation in the Chapter 5.

Chapter 2

Power System State Estimation via Globally Convergent Methods

2.1 State Estimation - Problem Formulation

Power system state estimation (PSSE) is an algorithm for determining the system state from a model of the power system network and redundant system measurements. Here we will describe a basic state estimation algorithm. The state estimation nonlinear measurement model is defined by:

$$z = h(x) + \epsilon$$

where:

z	m-dimensional measurement vector;
x	n-dimensional ($n < m$) state vector (of voltage magnitude and phase angle);
$h(x)$	nonlinear vector function relating measurements to states (m-vector);
ϵ	m-dimensional zero mean measurement error vector;
m	number of measurements;
n	number of state variables.

The problem is to determine the estimate x that best fits the measurement model. The static-state of an N bus electric power network is denoted by x, a vector of dimension $n = 2N - 1$, comprised of N bus voltages and $N - 1$ bus voltage angles. The state estimation problem can be formulated

as a minimization of the weighted least-squares (WLS) function problem

$$\min_{x\in\mathbb{R}^n} J(x) = \frac{1}{2}\left(z - h(x)\right)^T R^{-1}\left(z - h(x)\right) \tag{2.1}$$

or in terms of the residual vector

$$\min_{x\in\mathbb{R}^n} J(x) = \frac{1}{2} r^T R^{-1} r$$

where $r = z - h(x)$ is the residual vector; the nonlinear measurement function is defined as $h(x) : \mathbb{R}^n \to \mathbb{R}^m$

$$h(x) = \begin{pmatrix} h_1(x) \\ \vdots \\ h_m(x) \end{pmatrix} \in \mathbb{R}^m, \qquad z = \begin{pmatrix} z_1 \\ \vdots \\ z_m \end{pmatrix} \in \mathbb{R}^m$$

and R is a weighting matrix whose diagonal elements are often chosen as the measurement error variances, i.e.

$$R = E\{e \cdot e^T\} = \begin{pmatrix} \sigma_1^2 & & \\ & \ddots & \\ & & \sigma_m^2 \end{pmatrix} \in \mathbb{R}^{m\times m}$$

The problem defined by (2.1) is solved as an unconstrained minimization problem. An algorithm for such an unconstrained minimization problem is an iterative numerical procedure in which the objective function $J(x)$ is approximated usually by a quadratic model.

Efficient solution of unconstrained minimization problems relies heavily on some type of Newton's method. Newton's method has a central role in the development of numerical solution for unconstrained minimization problems. The type of Newton's method of most interest here is the Gauss-Newton method. There are two equivalent ways of defining it.

In the first approach, we linearize the nonlinear vector function $h(x)$ using Taylor series expansion

$$h(x + \Delta x) \approx h(x) + H(x)\Delta x$$

where the Jacobian matrix of dimension $m \times n$ is defined as:

$$H(x) = \begin{pmatrix} \frac{\partial h_1(x)}{\partial x_1} & \cdots & \frac{\partial h_1(x)}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial h_m(x)}{\partial x_1} & \cdots & \frac{\partial h_m(x)}{\partial x_n} \end{pmatrix} \in \mathbb{R}^{m\times n}$$

and then obtain the linearized least-squares objective function

$$J(\Delta x) = \frac{1}{2}\left(z - h(x) - H(x)\Delta x\right)^T R^{-1}\left(z - h(x) - H(x)\Delta x\right)$$

$$J(\Delta x) = \frac{1}{2}\left(r(x) - H(x)\Delta x\right)^T R^{-1}\left(r(x) - H(x)\Delta x\right).$$

The first-order necessary condition yields[2]

$$\frac{\partial J(\Delta x)}{\partial \Delta x} = -H^T R^{-1}\left(r - H\Delta x\right) = 0$$

which results in the well known normal equation

$$H^T R^{-1} H\Delta x = H^T R^{-1} r$$

In the second approach, given a starting point x_c, we construct a quadratic approximation m_c of the objective function $J(x_c)$ that matches the first and the second derivative values at that point

$$m_c(x_c + s) = J(x_c) + \nabla J^T(x_c)s + \frac{1}{2}s^T\nabla^2 J(x_c)s$$

Then we minimize the approximation (quadratic function) instead of the original objective function. Therefore, the first-order optimality condition is

$$\frac{\partial m_c}{\partial x} = 0$$

Finally, the normal equation is of the form

$$\nabla^2 J(x_k)s = -\nabla J(x_k)$$

where

$$\nabla J(x_c) = -H^T R^{-1} r$$

and the Hessian matrix $\nabla^2 J(x_c)$ is defined as:

$$\nabla^2 J(x_c) = H^T R^{-1} H + \underbrace{\sum_{i=1}^{m} r_i(x_c)\nabla^2 r_i}_{K}$$

The estimates are usually solved by Newton's method which computes the state corrections s at each iteration by solving:

$$\nabla^2 J(x^k)s = -\nabla J(x^k)$$

$$x^{k+1} = x^k + s$$

[2]We will write $H(x)$ as H in order to simplify the notation

for $k = 0, 1, 2 \ldots$ until convergence is attained.

In Newton's method, the Hessian matrix is computed exactly. K denotes the second-order information in $\nabla^2 J(x_c)$, which is often neglected in practice to avoid additional evaluation of m $n \times n$ Hessians. Moreover this term may produce an indefinite $\nabla^2 J$ which will ultimately lead to the Newton step being in a non-descent direction. Hence, the symmetric approximation of $\nabla^2 J(x)$ given by

$$\nabla^2 J(x_c) \approx H^T R^{-1} H$$

is used. $H^T R^{-1} H$ is called the Gauss-Newton Hessian. Consequently, by neglecting K in the method, we obtain the Gauss-Newton method as opposed to the full Newton's method. The difference between the two methods is that $\nabla^2 J(x)$ contains second order derivatives of $h(x)$ in the Newton method whereas these terms are not present in the Gauss-Newton method. Using the first approach, linearizing the nonlinear measurement function $h(x)$, the Gauss-Newton method is obtained right away, whereas using the second approach second order derivatives must be explicitly neglected. It has been shown by Van Amerongen in [92] that in practice, the impact of the second order derivatives is negligible when applied to PSSE. In what follows, we will restrict our attention to the Gauss-Newton method developed using the second approach.

When the Hessian $\nabla^2 J(x)$ (or its approximation) is nearly singular, applying the Newton method can produce a huge step that is often not in a descent direction. This can produce convergence failure. A descent direction for $J(x)$ at $x \in \mathbb{R}^n$ is a direction $s \in \mathbb{R}^n$ at which the condition $\nabla J(x)^T s < 0$ is satisfied. This condition will be tested in our algorithms as an indication of whether the method is heading in the right direction. One should always keep in mind the very important fact that the Newton step ($s^N = -\nabla^2 J(x)^{-1} \nabla J(x)$) is guaranteed to be a descent direction if and only if $\nabla^2 J(x)$ is positive definite. The Gauss-Newton step is $-(H^T R^{-1} H)^{-1} \nabla J(x)$, which is a descent direction as long as H is full-rank. We know that a stationary point is a minimizer if $\nabla^2 J(x)$ at that point is positive definite. In general in Newton's method, $\nabla^2 J(x)$ may not be positive definite during the iteration process and as a consequence Newton's method is not necessarily a descent method. Newton's algorithm is outlined in Alg. 1.

2.1.1 Orthogonal transformation

The state of the art in PSSE algorithms is either orthogonal factorization (QR factorization via Givens rotations) with ordering or the sparse tableau method (which is based on constrained

Algorithm 1 Newton's algorithm

```
given an initial x
until termination do
while ‖∇J(x)‖ > ε do
  evaluate ∇²J
  solve ∇²J s = −∇J(x)
  x ← x + s
end while
```

optimization). In this section we choose to cover QR factorization in detail particularly since we will use it in performance comparisons. Also globalized Newton's methods will be based on QR factorization.

The iterative equations using the Gauss-Newton method have the form:

$$\begin{aligned} &\left(H^T(x^k)R^{-1}H(x^k)\right)s = H^T(x^k)R^{-1}\left(z - h(x^k)\right) \\ &x^{k+1} = x^k + s \end{aligned} \tag{2.2}$$

Equations (2.2) are the so-called normal equations of the weighted least squares problem. In the above equation the term $G = H^T R^{-1} H$ is the so-called gain (information) matrix. Since the normal equations involve squaring the H matrix, the accuracy depends on the condition number of H,

$$\kappa_2(G) = \kappa_2^2(R^{-1/2}H)$$

While the normal equations can be solved using several methods, the numerically most stable method is the orthogonal transformation method [93] which will be used in comparison analysis as well as in developing the trust region method. Orthogonal factorization methods are very desirable because they do not magnify roundoff or any other kinds of errors due by avoiding building the gain matrix $H^T R^{-1} H$. The normal equations can be rewritten as:

$$H^T R^{-1/2} R^{-1/2} H s = H^T R^{-1/2} R^{-1/2} \left(z - h(x)\right)$$

Define

$$\begin{aligned} H_w &= R^{-1/2} H \\ r_w &= R^{-1/2}\left(z - h(x)\right). \end{aligned}$$

Then the normal equation can be written as

$$H_w^T H_w s = H_w^T r_w \tag{2.3}$$

Orthogonal transformation avoids squaring the H_w matrix by applying QR factorization to the weighted Jacobian H_w

$$H_w = \widehat{Q}^T \widehat{U}$$

where $\widehat{Q}$ is an orthogonal $(m \times m)$ matrix $(\widehat{Q}\widehat{Q}^T = I)$ and $\widehat{U}$ is an upper trapezoidal $(m \times n)$ matrix. Applying orthogonal transformation to (2.3) will result in

$$\widehat{U}^T \widehat{Q}\widehat{Q}^T \widehat{U} s = \widehat{U}^T \widehat{Q} r_w$$

$$\widehat{U} s = \widehat{Q} r_w$$

This equation can be solved in two steps:

$$\begin{aligned} y &= Q r_w \\ U s &= y \end{aligned}$$

where:

Q	$(n \times m)$ orthogonal matrix $\widehat{Q} = (Q^T \quad \bar{Q})^T$;
U	$(n \times n)$ upper triangular matrix $\widehat{U} = (U^T \quad 0)^T$;
r	residual vector $(r = z - h(h))$;
r_ω	weighted residual vector $(r_\omega = R^{-1/2} r)$.

The solution algorithm described above will be called Newton-QR throughout the rest of this dissertation.

2.1.2 Test Results

The Newton-QR algorithm based on Givens rotations has been implemented in MATLAB together with the Tinney 2 ordering scheme [93] and tested on standard IEEE system cases [90]. At this point, we will show that under normal assumptions on the measurement noise level the Newton-QR method performs reliably. We will also show cases where the Newton-QR is unable to find a solution. We will test the same cases with the trust-region method.

Besides the first-order necessary condition for optimization, or the fact that the gradient should vanish at the minimizer $(\nabla J(x) = 0)$, we observe the following parameters: the objective function $J(x)$, the norm of the step $\|s\|$, and the descent direction condition $\nabla J(x)^T s < 0$. As the iterative

process proceeds, the objective function and the step length should decrease. The descent direction criterion, which indicates whether the iteration process is heading in the right directions is straightforward to check by examining $\nabla J^T(x)s < 0$.

The first case consider the measurement set for the IEEE 14-bus network shown in Fig. A.1 on page 134 in the Appendix A. The performance is shown in Fig. 2.1 and in Table 2.1. The second case is the IEEE 30-bus system with the measurement set shown in Fig. A.2 page 136 of the Appendix A. Convergence of this case is shown in Fig. 2.2 and Table 2.2.

One can see that QR performs reliably in both cases. Moreover, Newton-QR was able to solve successfully many situations with a single topology error.

A single topology error in the IEEE 14-bus system can prevent convergence of the Newton-QR method. The measurement set described in Appendix A Fig. A.3 on page 137 is one such example. Behavior of the first-order necessary condition during the iteration process is shown in Fig. 2.3. Observing other parameters in Table 2.3, one can see that in iterations 1,4, and 5 the algorithm does not head in the descent direction: $\nabla J^T(x)s > 0$. Steps in the descent direction reduce both the objective function and the step norm.

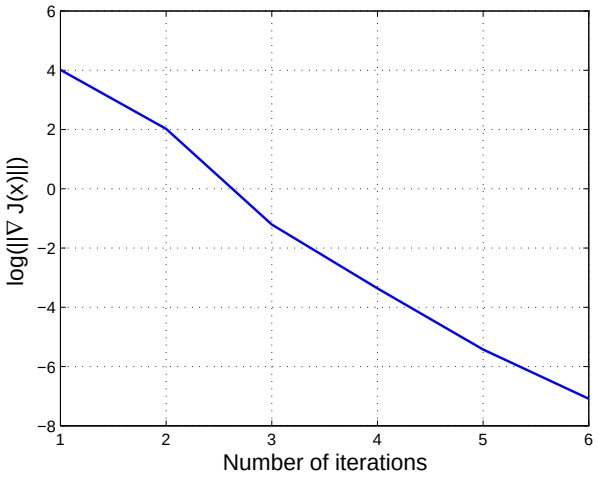

Figure 2.1: Convergence of the Newton-QR State Estimator for the IEEE 14-bus test case

2.1.3 Orthogonal transformation - Remarks

Although the numerical stability of solving the normal equation plays an important role in the overall algorithm, it should be regarded as just one of the goals in building robust state estimator.

Table 2.1: Newton-QR State Estimator applied to the IEEE 14-bus test case

# of iteration	$J(x)$	$\|s\|$
1	3933.85	0.9118
2	64.551	0.0771
3	12.26155015334864	$3.8156 \cdot 10^{-4}$
4	12.25774142035936	$1.1243 \cdot 10^{-6}$
5	12.25774142086011	$7.4052 \cdot 10^{-9}$
6	12.25774142357370	$5.9328 \cdot 10^{-11}$

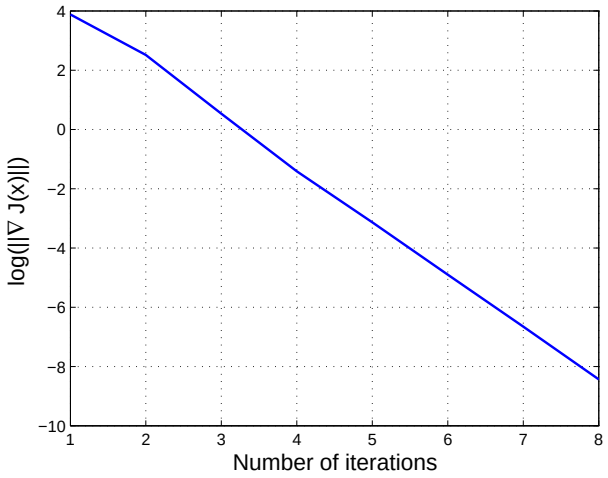

Figure 2.2: Convergence of the Newton-QR State Estimator for the IEEE 30-bus test case

Table 2.2: Newton-QR State Estimator applied to the IEEE 30-bus test case

# of iteration	$J(x)$	$\|s\|$
1	3750.167	1.5046
2	52.803	0.0840
3	7.91856500700121	$2.00 \cdot 10^{-3}$
4	7.89153387383260	$2.2860 \cdot 10^{-5}$
5	7.89154236506929	$2.3420 \cdot 10^{-7}$
6	7.89154217042462	$4.2913 \cdot 10^{-9}$
7	7.89154217376090	$7.0944 \cdot 10^{-11}$
8	7.89154217370184	$1.2465 \cdot 10^{-12}$

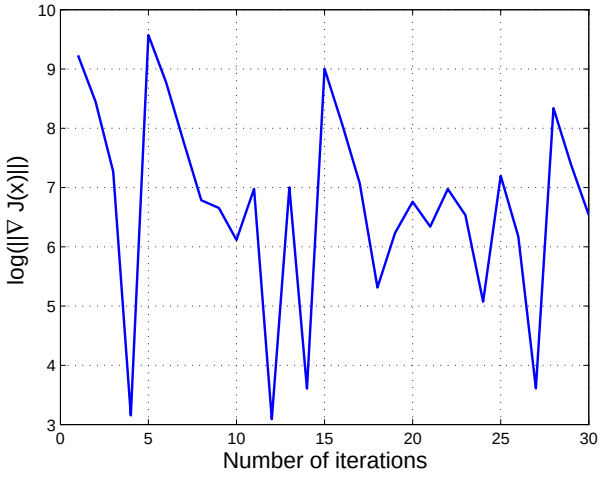

Figure 2.3: Newton-QR State Estimator applied to the IEEE 14-bus test case, the non-converging case

Table 2.3: Newton-QR State Estimator applied to the IEEE 14-bus test case, the non-converging case

# of iteration	$J(x)$	$\|s\|$	$\nabla J^T(x)s$
1	$3.8881 \cdot 10^3$	56.5160	$5.7048 \cdot 10^{10}$
2	$1.3418 \cdot 10^{10}$	19.2814	$-3.5647 \cdot 10^9$
3	$9.8887 \cdot 10^8$	12.2159	$-1.8289 \cdot 10^8$
4	$2.6988 \cdot 10^7$	10.2250	586.1503
5	15.8857	76.9153	$1.9210 \cdot 10^{11}$
6	$3.7584 \cdot 10^{10}$	24.4886	$-9.7947 \cdot 10^9$
7	$2.7010 \cdot 10^9$	14.0975	$-6.7856 \cdot 10^8$
$\vdots$	$\vdots$	$\vdots$	$\vdots$

If the gain matrix ∇J^2 of the normal equation is not positive definite, Newton's method may produce a huge step that is not in a descent direction, independent of the numerical conditioning of the matrix.

In the Gauss-Newton method the gain matrix $H^T R^{-1} H$ is always at least positive semi-definite. It fails to be positive definite if and only if H is rank deficient, in which case the gain matrix is singular, i.e., "infinitely" badly conditioned. The Gauss-Newton step can fail to be descent direction only if ill-conditioning results in excessive numerical error in the computed step.

2.2 Globally Convergent Methods - Introduction

The roots of the trust region algorithm methods lie in the pioneering work of Levenberg (1944) and Marquardt (1963) for nonlinear least squares problems. They first noticed that when the Hessian is not symmetric positive definite (SPD) in Newton's method, the method may not converge. Adding positive elements (the Levenberg-Marquardt parameter) to the diagonal was suggested. Although the criteria for selecting the Levenberg-Marquardt parameter were not theoretically sound, the idea was the foundation for work by Moré [59].

Newton's method works very well when the initial guess is near the solution. An overview of the Newton's method can be found in many references, e.g., Moré and Sorensen [61]. But what happens when we are not in the situation to provide a close initial guess? One idea is to augment Newton's method with "globalization". Globalization of Newton's method increases the likelihood of convergence from an arbitrary initial guess. Convergence cannot be guaranteed even with globalization.

Recall the properties of Newton's method:

1. the iterates may diverge if x_0 is not near the solution
2. as $x_n \to x_*$ convergence is usually quadratic (very fast)
3. each iteration requires evaluation and factorization of $\nabla^2 J$
4. convergence is only local
5. numerical difficulties may arise if $\nabla^2 J$ is ill-conditioned

The major strength of the Newton's method is its quadratic convergence near the solution. Before considering particular globalization methods, we describe the general structure of the globalized

Newton's method [96]

1. Begin with initial trial step (Newton step)
2. Test for adequate progress
3. Modify if necessary to get a new trial step; return to the test

This dissertation will present a new approach for solving power system state estimation based on a globally convergent modification of Newton's method using trust region methods. The objective is to provide a more reliable and robust state estimator, which can successfully cope with all kinds of errors (bad data, topological, parameter) faced in power system models.

There are two issues which a robust state estimation algorithm must be able to overcome. One of them is the numerical ill-conditioning problem which is solved quite successfully with QR factorization; the other is the convergence problem induced by data errors. When the system is ill-conditioned it will manifest itself in the form of slow convergence or failure to converge. Orthogonal transformation methods are more numerically stable than other methods. By applying them, the issue of ill-conditioning is mitigated. But even this algorithm can suffer from non-convergence in the face of faulty data. This dissertation is an attempt to remedy the second issue.

While there is no way to ensure that iterates will always converge to a solution of every problem, our motivation was to implement more "successful" methods to the power system state estimation problem in order to improve convergence in the presence of model errors. The approach we will present is well known in the field of numerical optimization. It consists of two globally convergent methods, the line search (backtracking) method and the trust region (restricted step) method. The trust region state estimator was first presented in [70]. We will provide a theoretical framework for both global methods and analyze them on standard IEEE network test cases. The backtracking method is included for completeness although it has not proved as reliable a global method as the trust region method, which will be our main concentration. Strong theoretical support as well as practical efficiency and robustness are the strong arguments supporting the trust region method for power system state estimation.

The standard technique for solving state estimation problems is to apply the Newton or Gauss-Newton method. While Newton's method has superior convergence properties when the starting point is near the solution, its disadvantage is possible convergence failure on problems that are very nonlinear. Whereas in the Gauss-Newton method very large residuals are the major issue that can prevent convergence. which are common in power system state estimation. All global algorithms

include calculation of the Newton step, because the strategy of the global methods is to apply the Newton step whenever possible. Certainly any global method will end up using Newton's method near the solution to exploit its fast local convergence rate.

The trust region method allows more control of the step calculation. The trust region is that region in the problem space in which we can trust that a quadratic model is an adequate model of the objective function. The measure of progress is the diameter of the trust region δ which is a controllable quantity; it can be expanded or reduced based upon how well the local model predicts the behavior of objective function.

2.2.1 The Backtracking (line search) Method

We will now explore the line-search way of modifying the Gauss-Newton step to obtain steps that satisfy acceptability criteria. The backtracking idea can be stated as: initially try the Gauss-Newton step; if a step is not acceptable, shorten it as necessary until an acceptable step is found. Line search iterative algorithms for finding a minimizer of $J(x)$ are of the form:

$$x^{k+1} = x^k + \theta s$$

for a given trial step s. The choice of θ ensures convergence criteria $J(x^{k+1}) < J(x^k)$; under section 2.2.3 on page 49, we will stress more strict convergence criteria which will hopefully force the sequence into a neighborhood of a local minimizer. The reduction with $\theta \in [\theta_{min}, \theta_{max}]$ is the so called "safeguarded" backtracking method [94].

- $\theta \leq \theta_{max}$ ensures that the backtracking (inner) loop will terminate with an acceptable step;
- $\theta \geq \theta_{max}$ ensures that steps will not be excessively small, producing poor convergence

The choice of θ_{min} and θ_{max} is arbitrary and problem dependent. We have used values suggested in [24] for practical implementations, $\theta_{min} = 0.1$ and $\theta_{max} = 0.5$. Our experience has been that the use of larger values for θ_{max} usually resulted in a larger number of inner loop iterations.

In choosing $\theta \in [\theta_{min}, \theta_{max}]$, we minimize a one dimensional quadratic/cubic interpolating polynomial $p(\theta)$ satisfying following constraints:

$$\begin{aligned} p(0) &= J(x^k) \\ p(1) &= J(x^k + s^k) \\ p'(0) &= \frac{d}{d\theta} J(x + \theta s)\Big|_{\theta=0} = \nabla J^T s \end{aligned}$$

The quadratic polynomial interpolating the above points is:

$$p(\theta) = \left[p(1) - p(0) - p'(0)\right]\theta^2 + p'(0)\theta + p(0).$$

The minimum of the above polynomial is

$$\theta = \frac{-p'(0)}{2\left[p(1) - p(0) - p'(0)\right]}.$$

If $J(x^k + s)$ does not satisfy the convergence criterion, subsequent reductions can be either quadratic or cubic interpolations. The proposed algorithm implements cubic subsequent interpolates; more detail can be found in [24].

In the inner loop, besides the one dimensional quadratic minimization, we need only to evaluate the objective function which does not require much computational effort.

Shortcomings of the backtracking approach include:

- While sometimes successful, the backtracking strategy has the disadvantage that it makes no further use of the n-dimensional quadratic model.
- Many step-length reductions may be required, entailing unproductive effort.
- The step may achieve relatively little reduction in the objective function, compared to other steps of the same length but in different directions.

As we saw, an unsatisfactory Newton step indicates that our quadratic model does not adequately model the objective function in a region containing the full Gauss-Newton step. The question arises: What is the region in which we can *trust* that the quadratic model is able to represent our objective function correctly? The trust region method is an attempt to answer this question.

2.2.2 Trust Region Method

As mentioned, our prime focus will be on trust region methods. The trust region method is a robust implementation of the algorithm whose origin lies in the work of Levenberg [53] and Marquardt [54]. A general trust-region-based algorithm is of the following form.

Minimize the local quadratic model m_c of objective function $J(x)$ over the region of radius δ centered at x_c

$$\begin{aligned} \min \quad & m_c(x_c + s) = J(x_c) + \nabla J^T(x_c)s + \frac{1}{2}s^T\nabla^2 J(x_c)s \\ \text{subject to:} \quad & \|s\| \leq \delta \end{aligned} \tag{2.4}$$

where:

m_c	quadratic model of the objective function reduction;
δ	trust region radius;
$\nabla J(x_c)$	gradient of objective function;
$\nabla^2 J(x_c)$	Hessian (or approximation);
$\|.\|$	2-norm throughout.

The trust region algorithm can be outlined as follows:

- choose a step s according to (2.4);
- check if sufficient reduction in objective function is achieved by the model;
- if the step is not acceptable, reduce δ and try again;
- once an acceptable step has been found, adjust δ for the next step.

The calculation of the step between iterates requires the solution of the above locally constrained minimization problem. Applying the Lagrange multiplier method to (2.4) will produce the following solution with μ as the multiplier corresponding to the trust region constraint:

$$\begin{aligned} &\left(\nabla^2 J + \mu I\right) s(\mu) = -\nabla J \\ &\text{such that} \quad \|s\| = \delta \end{aligned}$$

Considering the Gauss-Newton case the above equation will have form:

$$\begin{aligned} &\left(H^T R^{-1} H + \mu I\right) s(\mu) = H^T R^{-1}(z - h(x)) \\ &\text{such that} \quad \|s(\mu)\| = \delta \end{aligned} \tag{2.5}$$

The first equation in (2.5) can be rewritten in following form:

$$\begin{pmatrix} H^T R^{-1/2} & \mu^{1/2} I \end{pmatrix} \cdot \begin{pmatrix} R^{-1/2} H \\ \mu^{1/2} I \end{pmatrix} s(\mu) = H^T R^{-1} r \tag{2.6}$$

The solution process applies QR factorization to the following matrix:

$$\begin{pmatrix} R^{-1/2} H \\ \mu^{1/2} I \end{pmatrix} = Q_k^T U_k. \tag{2.7}$$

Since we already have factored the upper block matrix, it is only necessary to process the additional diagonal elements. This property is particularly appealing if we need several iterations. We calculate the trust region step in a very similar way to the Gauss-Newton step

$$U_k^T U_k s(\mu) = U_k^T Q r_w. \tag{2.8}$$

The right hand side is calculated once in the outer Newton iteration, while U_k is calculated by rotating $\mu^{1/2} I$ into U. Finally, the step is calculated by forward/backward substitution.

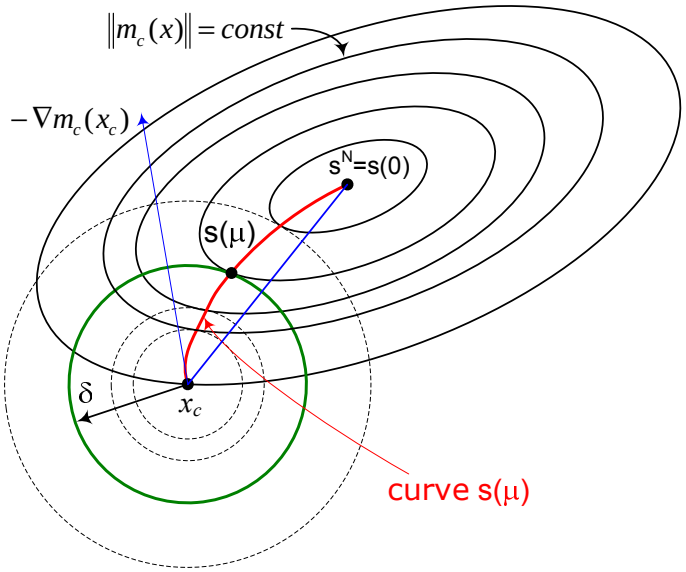

Figure 2.4: The curve $s(\mu)$

The $s(\mu)$-curve $\{s(\mu) : 0 \le \mu < \infty\}$ is defined by

$$s(\mu) = \left[H^T R^{-1} H + \mu I\right]^{-1} H^T R^{-1} r.$$

As shown in Fig. 2.5, it traces out a differentiable curve of the trust region steps. $\|s(\mu)\|$ is monotone decreasing in μ, with

- $\lim_{\mu \to 0} \|s(\mu)\| = \|s^N\|$
- $\lim_{\mu \to \infty} \|s(\mu)\| = 0$

A fundamental practical difficulty is that due to the nonlinear constraint, there is no direct method for solving equation (2.5); thus we cannot determine exactly an $s(\mu)$ such that $\|s(\mu)\| = \delta$. Therefore our task is to determine an adequate approximation at a reasonable cost. We will use

the approximate method ("hook" step) suggested in [24], [59]. The idea can be outlined: determine $s = s(\mu)$ exactly for μ such that $\|s(\mu)\|$ is approximately δ. One implementation formulation would be to find an approximate solution to the following scalar nonlinear equation for some strictly positive value of μ:

$$\Phi(\mu) = \|s(\mu)\| - \delta = 0. \tag{2.9}$$

Since there is no need for great accuracy, we will use the recommendation in [24] (Fig. 2.5) to terminate iterations as soon as

$$\frac{3}{4}\delta \leq \|s(\mu)\| \leq \frac{3}{2}\delta$$

although some other values are possible. The key point is that it does not influence the number of

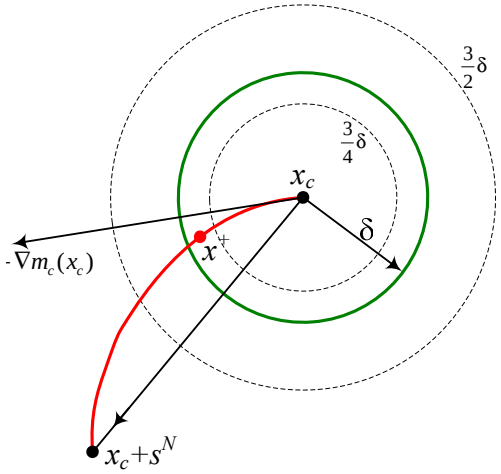

Figure 2.5: Calculation of trust region step

iterations needed to solve (2.9), usually no more than 2. Solving $\Phi(\mu) = 0$ is a scalar zero-finding problem in μ. Although one might first consider using Newton's method to solve this problem, it can be easily shown that it may not perform very well due to the structure of $\Phi(\mu)$. As suggested in [61], $\|s(\mu)\|^2$ can be written in the form

$$\|s(\mu)\|^2 = \sum_{i=1}^{n} \frac{\gamma_i}{(\lambda_i + \mu)^2}$$

where $\lambda_1, \ldots, \lambda_n$ is the spectrum of $H^T R^{-1} H$. Since $\|s(\mu)\|^2$ is a rational function in μ and has second order poles at $-\lambda_1, \ldots, -\lambda_n$, Newton's method tends to perform poorly when the solution is near $-\lambda_i$, for $i = 1, \ldots, n$. However, this does not present a problem in our case since $\mu \geq 0$ and

$\lambda_i > 0$ for each i. Fig. 2.6 shows the function $\|s(\mu)\|^2$ with the assumption of symmetric positive definite Hessian $H^T R^{-1} H$ with eigenvalues $\lambda_1 > \lambda_2 > \cdots > \lambda_n$.

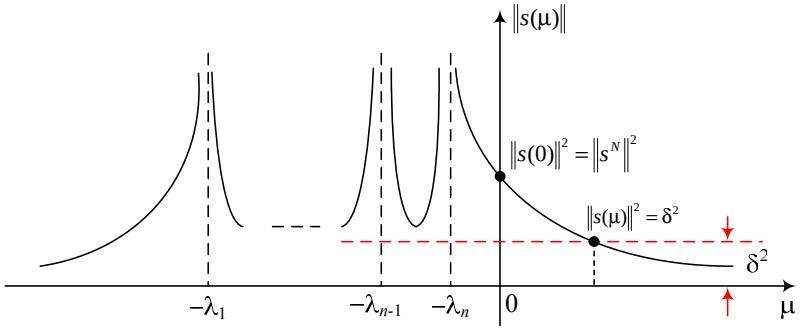

Figure 2.6: Sketch of $\|s(\mu)\|^2$

Several possible implementations have been proposed for solution of (2.9). We will discuss the one presented in [24] which suggests using a local model of the form according to the structure of the previous equation:

$$q_c(\mu) = \frac{\alpha_c}{\beta_c + \mu} - \delta_c$$

with current values of α_c and β_c that are changing in the inner iterations and calculated easily from the following initial conditions:

$$\begin{aligned} q_c(\mu_c) &= \Phi(\mu_c) \\ q_c'(\mu_c) &= \Phi'(\mu_c). \end{aligned}$$

Therefore, μ is calculated so that $q_c(\mu) = 0$ which will ultimately result in the following iterative process:

$$\mu_{c+1} = \mu_c + \frac{\|s(\mu)\|}{\delta_c} \cdot \frac{\Phi(\mu_c)}{\Phi'(\mu_c)}.$$

The above iterative process must be safeguarded in order to converge; we specify upper and lower limits according to [24], [59]. Since each evaluation of $\|s(\mu)\|$ requires the solution of a system of linear equations (2.8), it is crucial to solve this problem in a few iterations. A very important property is that the number of iterations required to determine an acceptable value of μ is very small (one to two iterations) because the iteration process itself is based upon the rational structure of Φ.

2.2.3 Criteria for Global Convergence

One of the serious drawbacks of Newton's method is that in its pure form, it does not necessarily produce a descent direction. Therefore, it is crucial to distinguish between "successful" and "unsuccessful" iterates. When applying the pure Newton's method, as seen in Alg. 1, we just assume that the sequence of iterates will ultimately converge to the solution, without any testing. It will be shown that, in the presence of certain network topology errors, Newton's method does not converge. In developing both the backtracking and the trust region algorithm, we will introduce a criterion for global convergence, i.e., a step-acceptance rule, as a criterion for the sequence of iterates to converge to a solution.

There are several alternative criteria for a step-acceptance rule. A simple condition that requires $J(x^{k+1}) < J(x^k)$ does not guarantee that the sequence of iterates will converge to a minimizer. There are examples in [24] that show how the condition $J(x^{k+1}) < J(x^k)$ can be satisfied but the iterates still fail to converge to a minimizer. Therefore, we need stronger convergence conditions. The most widely used rules are:

1. Goldstein-Armijo
2. `ared/pred`

The Goldstein-Armijo conditions are defined as follows: For $0 < \alpha < \beta < 1$ and a descent direction s, (i.e. $s \in \mathbb{R}^n$ is a descent direction for $J(x)$ at $x \in \mathbb{R}^n$ if $\nabla J(x)^T s < 0$)

$$J(x+s) \leq J(x) + \alpha \nabla J(x)^T s \qquad \text{alpha condition}$$
$$\nabla J(x+s)^T s \geq \beta \nabla J(x)^T s \qquad \text{beta condition}$$

The condition $0 < \alpha < \beta < 1$ ensures that there exists a step that satisfies these conditions. In a global optimization framework, the Goldstein-Armijo conditions were suggested in [24]. When applied to our problem, the results were not encouraging. Many times the iteration process stagnated.

The second test, and the one we have used, is known as the `ared/pred` criterion. This criterion requires:

$$ared \geq t \cdot pred$$
$$ared = J(x_c) - J(x_c + s)$$
$$pred = J(x_c) - m_c(x_c + s) = -\nabla J(x_c)^T s - \frac{1}{2} s^T H^T R^{-1} H s$$

where:

`ared`	actual reduction in $J(x)$;
`pred`	reduction in $J(x)$ "predicted" by the local quadratic model $m_c(x)$ of $J(x)$;
$t \in (0, 1)$	usually t is very small so a step could be accepted if there is minimal (but still adequate) progress [94].

`ared/pred` criteria for the trust-region algorithm were suggested in [60]. `ared/pred` criteria were much more reliable when applied to our problem. Hence we decided to implement them in our algorithm.

The step-acceptance rule determines whether the trial step is accepted or not. If the trial step is unacceptable, the trust region will be reduced in an inner loop and minimization of the same quadratic function performed on a smaller trust region radius. The reduction factor θ is determined by minimizing the one-dimensional quadratic model interpolated between $J(x_c)$ and $J(x_c + s)$. We do not want to decrease the trust region too much; therefore, there will be imposed lower θ_{min} and upper θ_{max} limits on the reduction factor. While values for θ_{min} and θ_{max} can be chosen arbitrarily, often (as suggested in [24]) the choices are 0.1 and 0.5 respectively. After the new trust region is determined the algorithm returns to the approximate solution of the locally constrained minimization problem.

According to the above criteria, specific rules will be designed for revising and maintaining the trust region radius δ during the iteration process in an outer loop.

There are three cases of interest. The first is when there is excellent agreement between $J(x)$ and local quadratic model, the second case is when the agreement is acceptable, and the third case is when the agreement is poor.

Updating of the trust region radius is done as follows:

$$\delta \leftarrow 2\delta \quad \text{if} \quad \frac{ared}{pred} \geq u$$

$$\delta \leftarrow \delta \quad \text{if} \quad v \leq \frac{ared}{pred} < u$$

$$\delta \leftarrow \delta/2 \quad \text{if} \quad \frac{ared}{pred} < v$$

Values of u and v recommended in [24] are $u = 0.75$ and $v = 0.1$. Other values may be considered as well.

2.2.4 The backtracking algorithm

The backtracking algorithm [94] is outlined in Algorithm 2.

Algorithm 2 Backtracking algorithm

given $t \in (0,1)$ and $0 < \theta_{min} < \theta_{max} < 1$

evaluate $J(x)$, $\nabla J(x)$

Iterate

while $\|\nabla J(x)\| > \epsilon$ **do**

  calculate s (Gauss-Newton step)

  evaluate $J(x+s)$

  while $ared < t \cdot pred$ **do**

    choose $\theta \in [\theta_{min}, \theta_{max}]$

    update $s \leftarrow \theta s$

    re-evaluate $J(x+s)$

  end while

  update $x \leftarrow x+s$, and $J(x) \leftarrow J(x+s)$

  evaluate $\nabla J(x+s)$ and update $\nabla J(x) \leftarrow \nabla J(x+s)$

end while

As we initially mentioned in section 2.2.1 on page 43, our backtracking algorithm is "safe-guarded" so that it terminates in a finite number of steps. Simulation results have shown that the number of inner (backtracking) iterations was usually very small (2-3) and never exceeded 6.

2.2.5 Trust Region Algorithm

The basic trust region algorithm is outlined in Algorithm 3

2.2.6 Simulation Results

In practice there are several reasons for the failure of the state estimator, even when it is based on the orthogonal transformation method. Among the reasons for convergence failure are very large measurement errors, parameter errors and/or topology errors. We have investigated the effect of topology errors on convergence. These types of errors are very severe because they affect several local measurement residuals. Residuals produced by these errors can cause the state estimator to fail to converge to a solution even when one exists.

Algorithm 3 Trust Region Algorithm

given $t \in (0,1)$, $0 < \theta_{min} < \theta_{max} < 1$, $0 < v < u < 1$ and $\delta > 0$

evaluate $J(x)$, $\nabla J(x)$

Iterate

while $\|\nabla J(x)\| > \epsilon$ **do**

calculate s

$$s(\mu) = -\left(H^T R^{-1} H + \mu I\right)^{-1} H^T R^{-1}(z - h(x))$$

such that $\|s(\mu)\| = \delta$

evaluate $J(x+s)$, $m_c(x+s)$

while $ared < t \cdot pred$ **do**

choose $\theta \in [\theta_{min}, \theta_{max}]$

update $\delta \leftarrow \theta\delta$

calculate a new s

$$s(\mu) = -\left(H^T R^{-1} H + \mu I\right)^{-1} H^T R^{-1}(z - h(x))$$

such that $\|s(\mu)\| = \delta$

re-evaluate $J(x+s)$, $m_c(x+s)$

end while

update $x \leftarrow x + s$, and $J(x) \leftarrow J(x+s)$

if $ared \geq u \cdot pred$ **then**

$\delta \leftarrow 2\delta$

else if $ared < v \cdot pred$ **then**

$\delta \leftarrow \delta/2$

else

same δ

end if

evaluate $\nabla J(x+s)$ and update $\nabla J(x) \leftarrow \nabla J(x+s)$

end while

The trust region and backtracking algorithms were tested on several IEEE test systems. The Gauss-Newton algorithm presented in the convergence comparison is based on the orthogonal transformation method (QR factorization). The first scenario is a very common case in which we already saw the non-convergence caused by a single topology error in Fig. 2.3. This case is investigated on the IEEE 14-bus network with measurement set presented in Fig. A.3 on page 137 in Appendix. A single topology error (line 12 out) was simulated. A dashed line branch denotes a topology error in which we assume that the line is out when it is actually in.

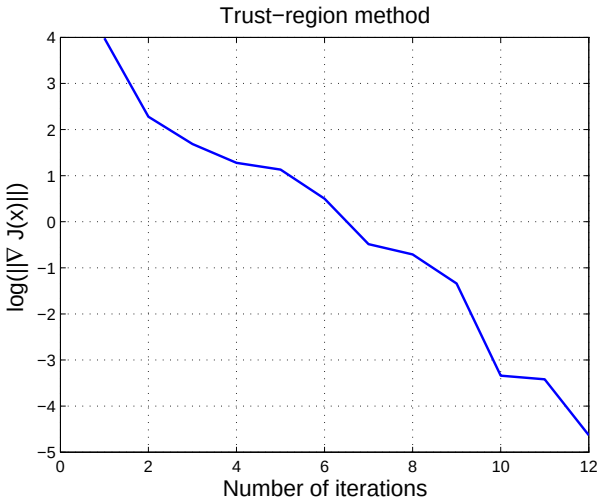

Figure 2.7: Convergence of the Trust Region State Estimator for the IEEE 14-bus test case

First we will present successful solution by the trust region method. Results are shown in Fig. 2.7 and in Table 2.4 and Table 2.5. It has to be pointed out that there is no need that $\nabla J(x)$ converge very accurately to obtain meaningful results. One has to keep in mind that voltage magnitude is estimated in per-unit and that accuracy of the gradient $\nabla J(x)$ of 10^{-4} is sufficient for practical purposes. Besides the parameters already discussed, we added the number of inner iterations to Table 2.4, to indicate the cost of the method. One can notice that the method is constantly in descent direction ($\nabla J^T(x)s < 0$) as opposed to Newton-QR presented in Fig. 2.3. The number of inner iterations is not excessively large, and the method is performing plain Newton's iterations near the solution.

In Table 2.5 we compare estimates obtained by the trust region method with the exact solution which was provided by the test case in [90]. Branch 12, that is modeled out when it is actually in, is connected between buses 6 and 12, as seen in Fig. A.3 in Appendix A. It is to be expected

Table 2.4: The IEEE 14-bus test case: Trust region method iteration process

# of iter.	$J(x)$	$\|s\|$	$\nabla J^T(x)s$	**# of inner iter.**
1	$3.8881 \cdot 10^3$	0.8949	$-7.7652 \cdot 10^3$	3
2	50.2372	2.3001	-95.5471	2
3	5.6373	0.1302	-6.2520	3
4	2.69719653128881	0.0736	-0.3752	2
5	2.52886228627499	0.1467	-0.0397	1
6	2.51570907143946	0.0527	-0.0125	2
7	2.50974212747101	0.0175	$-8.4643 \cdot 10^{-4}$	2
8	2.50930135450266	0.0158	$-4.8502 \cdot 10^{-5}$	2
9	2.50927238291790	0.0076	$-9.5334 \cdot 10^{-6}$	2
10	2.50926775714393	$7.6225 \cdot 10^{-4}$	$-1.9599 \cdot 10^{-7}$	2
11	2.50926768735649	$7.6223 \cdot 10^{-4}$	$-5.2979 \cdot 10^{-8}$	1
12	2.50926766478838	$1.1572 \cdot 10^{-4}$	$-1.5051 \cdot 10^{-9}$	0

that voltages at those two busses are not estimated very accurately. While the voltage at bus 6 is estimated relatively close to the solution one can see that the voltage phasor of bus 12 has an excessively low and unrealistic value, indicating either measurement or topology error.

Besides estimating the state of the system, SE is able to identify bad data. Bad data identification is the process of identifying noise corrupted measurements. and is conducted by performing normalized residual test (r^N-test). At this point we will state the basic idea behind bad data analysis, we refer the interested reader to either [1] or [57], where detailed treatment of bad data analysis can be found. Let $\widehat{z} = h(\widehat{x})$ denote an estimate of the measurement vector z, where $\widehat{x}$ is an estimate of the state vector. The covariance matrix of the estimate of the measurement vector $\widehat{z}$ is

$$R_{\widehat{z}} = H \left(H^T R^{-1} H\right)^{-1} H^T$$

The difference between the real measurement and the estimated measurement covariance matrix

$$W = R - R_{\widehat{z}}$$

is the measurement residual covariance matrix. Therefore, the measurement residuals are normal random variables with zero mean and covariance matrix W ($r \sim N(0, W)$). The normalized residuals for measurement i can be defined as

$$r_i^N = \frac{r_i}{\sqrt{W_{ii}}}$$

The normalized residual vector r^N is Gaussian random variable with a zero mean and unit variance ($r^N \sim N(0, 1)$). Thus existence of the bad data is identified by comparing the normalized residual

Table 2.5: State Estimates of the IEEE 14-bus test case solved by the Trust Region Method

	Solution		**Estimates**	
bus #	V **[pu]**	θ [°]	V **[pu]**	θ [°]
1	1.0603	0.0000	1.0602	0.0000
2	1.0451	-4.9754	1.0450	-4.9760
3	1.0094	-12.7012	1.0093	-12.7010
4	1.0192	-10.3265	1.0191	-10.3267
5	1.0202	-8.7753	1.0201	-8.7792
6	1.0697	-14.2225	1.0724	-13.9884
7	1.0621	-13.3637	1.0621	-13.3617
8	1.0902	-13.3537	1.0901	-13.3517
9	1.0561	-14.9419	1.0559	-14.9034
10	1.0509	-15.0965	1.0513	-15.0673
11	1.0568	-14.7905	1.0574	-14.7506
12	1.0548	-15.0721	0.4400	13.8980
13	1.0501	-15.1698	1.0500	-15.0137
14	1.0357	-16.0368	1.0361	-15.8908

against an appropriate threshold. In our test case, the conducted normalized residual test is depicted in Table 2.6 where suspicious measurement residuals are printed in red, The network placement of the suspicious measurements is presented in Fig. 2.8 were is one can see that all of them are in the vicinity of topology error. Therefore SE will detect topology error can be identified although indirectly. Combining result in the Table 2.5 and Table 2.6 conclusion is that topology error produce strange voltage values in incident nodes where and measurements in close proximity have very large residuals.

A convergence comparison for Gauss-Newton, backtracking and trust region method is plotted as $\log\left(\|\nabla J(x_c)\|\right)$ vs. the number of iterations for each test case. Fig. 2.9 compares the convergence of the three methods for the IEEE 14-bus case. It is seen that the Gauss-Newton method exhibits oscillatory nonconvergence and that the backtracking method stalls, failing to reach an acceptable solution. The trust region method converges for this test case.

The next case is the IEEE 30-bus network with the measurement set shown in Fig. A.4 on page 138 in the Appendix A. There are three topology errors indicated.

Fig. 2.10 compares the convergence of the three methods for the IEEE 30-bus case. The convergence behavior for each of the methods is similar to that for the IEEE 14-bus case.

The last case is the IEEE 118-bus network with ten topology errors. One might argue that ten topology errors are rare in practice; however, such a situation can occur with cascading failures. In such situations, a reliable state estimator is crucial. Convergence properties are shown on Fig. 2.11.

Table 2.6: The IEEE 14-bus test case: Normalized Residual Test

measurement #	**type**	**bus**	**line**	r	r^N
1	5	1	0	-0.0002	-0.0258
2	1	0	1	-0.0001	-0.0062
3	3	0	-2	0.0005	0.0224
4	2	2	0	-0.0001	-0.0099
5	4	2	0	-0.0002	-0.0409
6	1	0	5	-0.0003	-0.0114
7	3	0	5	0.0004	0.0145
8	1	0	-3	-0.0009	-0.0531
9	5	3	0	0.0007	0.0828
10	4	3	0	-0.0008	-0.0888
11	1	0	-4	0.0000	0.0006
12	3	0	-4	-0.0004	-0.0144
13	3	0	-7	0.0000	0.0020
14	2	4	0	-0.0001	-0.0275
15	1	0	-8	-0.0004	-0.0350
16	3	0	-8	0.0002	0.0174
17	1	0	14	0.0000	0.0293
18	4	8	0	0.0002	0.0192
19	5	8	0	-0.0001	-0.0192
20	1	0	-9	-0.0012	-0.0386
21	3	0	-9	0.0005	0.0203
22	5	9	0	0.0001	0.0212
23	2	9	0	-0.0002	-0.0308
24	3	0	-17	0.0023	0.0966
25	4	14	0	-0.0013	-0.0686
26	1	0	20	0.0002	0.0326
27	3	0	20	0.0001	0.0058
28	5	13	0	0.0000	-0.0083
29	1	0	19	0.0388	2.2002
30	3	0	-12	-0.0243	-0.7693
31	1	0	13	-0.0185	-1.8223
32	1	0	-11	0.0198	1.0751
33	3	0	-11	0.0012	0.0726
34	5	11	0	-0.0004	-0.0575
35	2	10	0	0.0006	0.0489
36	3	0	18	0.0000	-0.0488
37	1	0	10	0.0177	1.2491
38	5	5	0	-0.0001	-0.0103
39	2	6	0	0.0186	1.8375
40	4	6	0	0.0005	0.0883
41	1	0	-16	-0.0010	-0.0476
42	2	12	0	-0.0388	-1.8944

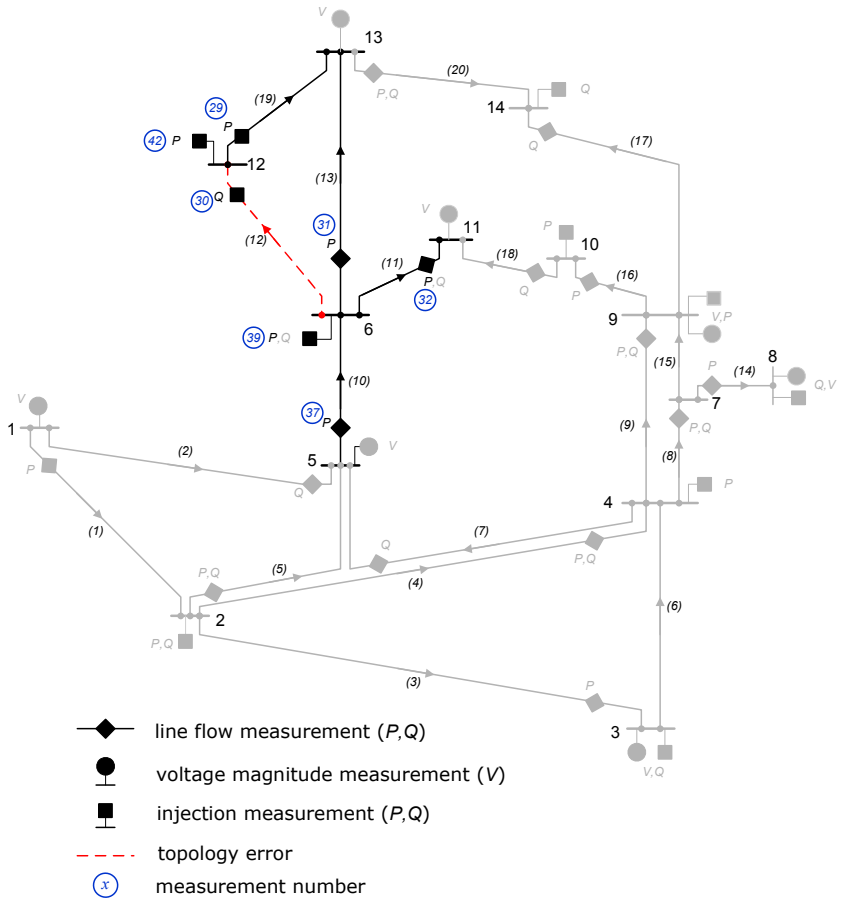

Figure 2.8: IEEE 14-bus test case - Topology Error Identification

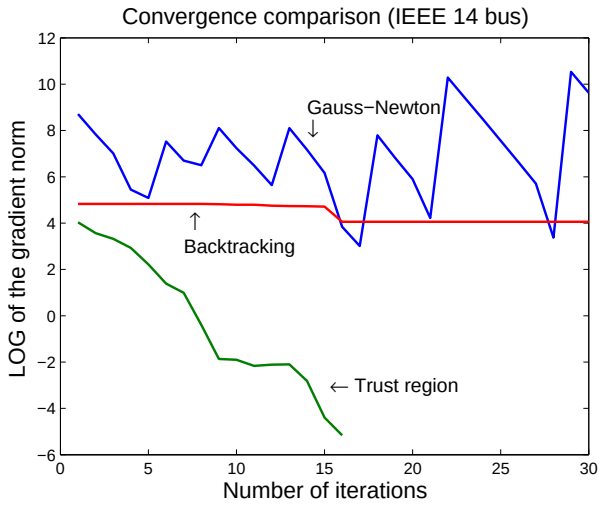

Figure 2.9: Convergence comparison for the IEEE 14-bus network with a single topology error.

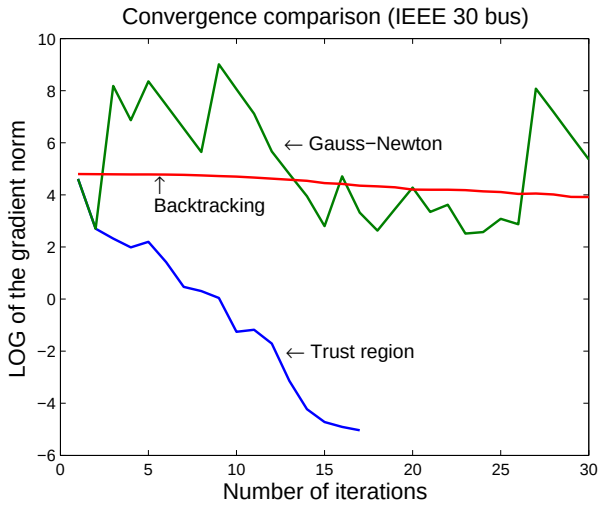

Figure 2.10: Convergence comparison for the IEEE 30-bus network with three topology errors.

It can be noticed that in the trust region method the number of iterations varies slightly with network size. The rate of convergence of the backtracking method in all three cases barely decreased.

In order to show that the rate of convergence is not always as poor as shown in the last three cases, we examined another test case. The IEEE 30-bus network was considered with four topology errors, which cause the Gauss-Newton algorithm to diverge (Fig. 2.12).

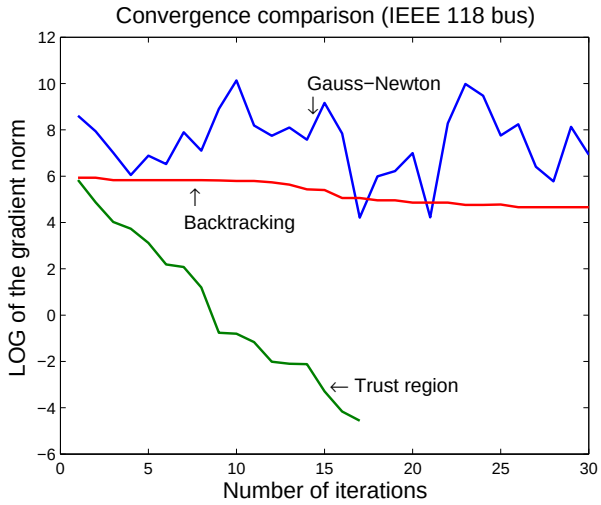

Figure 2.11: Convergence comparison for the IEEE 118-bus network with ten topology errors.

The backtracking state estimator was run for three cases which differ only in measurement noise

(the noise was modeled as Gaussian). One notices that even small changes of the order of magnitude of noise can significantly impact the rate of convergence of the backtracking algorithm.

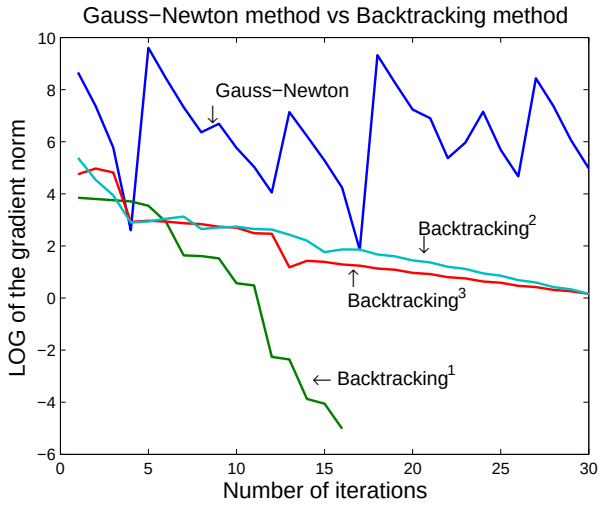

Figure 2.12: Convergence comparison of the Gauss-Newton versus Backtracking method for the IEEE 30-bus network with four topology errors.

The results illustrate the main advantage of global methods, and demonstrate that trust region methods can successfully cope with topology error cases where neither the Gauss-Newton method nor the backtracking method are able to reach the solution.

In terms of computational time, the backtracking method is comparable to the Gauss-Newton method since the inner loop requires one-dimensional minimization which is fast to obtain even for multiple step reductions. For the trust region method it is harder to give an exact analysis. The reason is that one can not predict the number of inner iterations needed per outer iteration. Our experience has shown that the number of inner iterations for the first six to eight outer iterations is usually two, while in some rare cases we encountered up to six inner loop iterations. For each inner iteration, the factorization (2.7) and solution of the linear system (2.8) are needed. A comparison of computational time for the three methods was not possible since neither the Gauss-Newton nor the backtracking method converged for our test cases.

In complex situations, like those arising from topology errors, the trust region method turns out to be very successful and it is rare that a solution is not found. As in any other numerical method, the trust region method has places of potential difficulty or break-down caused by finite precision. In the trust region algorithm, one of the most dangerous stages we faced and referenced in [20] is, perhaps surprisingly, near convergence. The problem arises due to floating point calculation of the

term `ared/pred`. When both differences are close to the machine precision, calculating `ared/pred` may result in a wrong sign (i.e., instead 1 we can easily get -1). Error in computation will result in further reduction of the trust region radius, which causes even more pronounced cancellation. As a result the algorithm will produce unsuccessful iterations and the convergence curve will stagnate close to the solution. A practical recommendation [20] is to treat `ared/pred=1` whenever absolute values of both `ared` and `pred` are smaller than some threshold value.

2.2.7 Conclusion

Computation of the state estimate for large networks, in the presence of bad measurement data, parameter, and/or topology errors requires a robust algorithm. Recent blackouts demonstrate the requirement to build more reliable state estimators. In this chapter, we focus on the robust implementation of a state estimator based on trust region methods. The trust-region method is a descent method, meaning that a trial point $x^{k+1} = x^k + s$ is accepted only if fulfill the step acceptance criterion. This approach to the problem leads to a very reliable situation, but one which is somewhat more computationally involved. The trust region method-based state estimator was found to be very reliable under severe conditions. This enhanced reliability justifies the additional time and computational effort required for its execution.

2.2.8 Historical Notes and Background

As we already mentioned, the foundation of the trust region method lies in the work of Levenberg [53] and later Marquardt, who surprisingly found out about Levenberg's work during the revision of his paper [54]. In [54] Marquardt defined the trust region method although he used the name *maximum neighborhood method*. His procedure was: Minimize the objective function (J) in the neighborhood over which the Taylor series approximation is an adequate representation of the nonlinear objective function.

He emphasized that any improved method will in some sense interpolate between a steepest-descent step s^g and a Newton step s^N such that the objective function of the least squares is reduced, $J^{(k+1)} < J^{(k)}$, where

$$s^g = -\nabla J$$

$$\nabla^2 J s^N = -\nabla J$$

In this approach, direction and step size are determined simultaneously instead of choosing a direction and trying to shorten it until an acceptable step is found. The theoretical basis of the *maximum neighborhood method* is contained in the theorem:

Theorem. *Let $\mu \geq 0$ be arbitrary and let s satisfy the equation*

$$\left(\nabla^2 J + \mu I\right) s = -\nabla J$$

Then s minimizes J on the sphere whose radius δ satisfies

$$\|s\|^2 = \delta^2$$

Marquardt suggested optimum interpolation between the Newton and steepest-descent step, although he did not suggest how to find μ such that $\|s\|^2 = \delta^2$. He said "some form of the trial and error is required to find a value of μ".

At that point the foundation for the trust region method was laid down, although a robust implementation was missing. The generalization of a result due to Marquardt and the computational aspect of the trust-region method (although the name trust-region was still not used), again considered for the least squares problem, was fully discussed by Moré in [59]. Moré called the method robust implementation of the Levenberg-Marquardt algorithm. Many parts of our algorithm that we used were proposed in [59]. Moré called the parameter μ the Levenberg-Marquardt parameter and presented an efficient procedure for finding μ such that $\|s(\mu)\|^2 = \delta$. Moré based the choice of trust-region radius δ on the actual and the predicted reduction of the objective function. His work included numerical and convergence results.

Since the early '80s, there has been an explosion in the research on trust-region methods. A number of state-of-the-art papers have been appeared since then. The name *trust region* was first used by Dennis. A thorough analysis of the locally constrained quadratic minimization problem defined by (2.4) that arises as a subproblem in the trust-region Newton iteration is given in [83]. This reference covers both the theoretical nature and possible implementation of the locally constrained model problem. Convergence criteria based on the `ared/pred` condition was suggested in this work as well as in the work of Shultz *et. al.* in [79].

The book by Dennis and Schnabel [24] is an invaluable source in this research. We highly recommend this reference to understand the ideas behind Newton's method in general and trust region methods in particular. The subject was covered thoroughly in the first book about trust region methods by Conn, Gould and Toint [20].

Besides the "hook" step approach for approximately finding $\|s(\mu)\| = \delta$, another commonly used approach is the dogleg approach. The idea behind the dogleg approach suggested in [24] is: Determine s such that $\|s\| = \delta$ exactly on a curve that approximates the $s(\mu)$-curve. The dogleg curve, shown in Fig. 2.13 is the polygonal curve connecting $s = 0$, $s = s^{SD}$ and $s = s^{N}$.

$$\min \quad m_c(x_c + s) = J(x_c) + \nabla J^T(x_c)s + \frac{1}{2}s^T\nabla^2 J(x_c)s$$

$$s = -\lambda \nabla J(x_c)$$

$$\min_{\lambda \in \mathbb{R}} \quad m_c\left(x_c - \lambda \nabla J(x_c)\right)$$

$$\lambda = -\frac{\|\nabla J(x_c)\|^2}{\nabla J(x_c)^T \nabla^2 J(x_c) \nabla J(x_c)}$$

which gives the step in steepest-descent direction

$$s^{SD} = x_c - \frac{\|\nabla J(x_c)\|^2}{\nabla J(x_c)^T \nabla^2 J(x_c) \nabla J(x_c)}$$

Globally convergent methods in general, and trust-region methods in particular, were not applied to PSSE before this research. We first introduced trust region methods to the power community in [70] with a more comprehensive treatment given in [71].

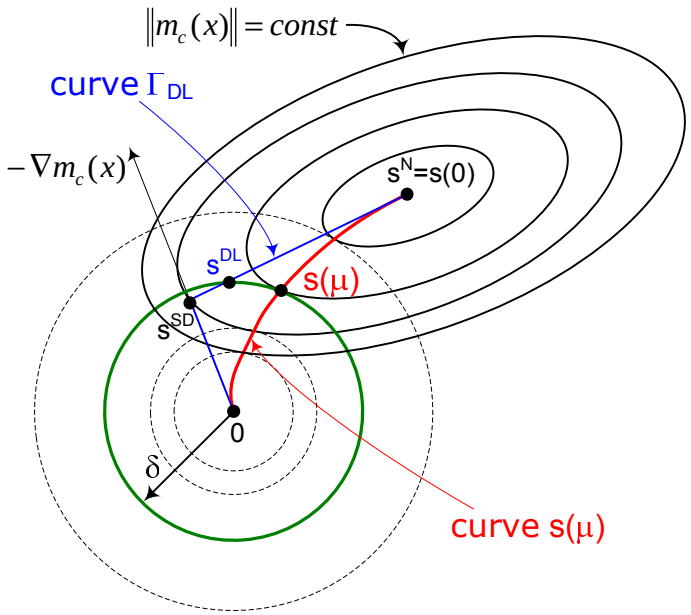

Figure 2.13: The dogleg (Γ_{DL}) curve

Chapter 3

Newton-Krylov Methods in Power System State Estimation

3.1 Introduction

As a real-time application in modern power systems, the state estimator is required to be numerically robust and fast. We stressed numerically robust techniques in the previous chapter. The general conclusion is that while numerically very stable, a QR factorization-based state estimator can not successfully handle severe cases resulting from uncertainty in the system. In these situation a trust-region method-based state estimator provides reliable solution. Under normal system conditions, QR serves as a reliable state estimator. This chapter studies aspects of iterative methods and their implementation relative to state estimation. The question we try to answer in this chapter is whether a faster or numerically less expensive solution method with a level of reliability comparable to QR exists in the pool of Krylov subspace methods.

Over the years, many different algorithms have been proposed for efficient solution of the power system state estimation problem. When it comes to reliability and robustness of the solution, QR factorization-based state estimator is the algorithm of choice. Among favorable properties that distinguish solving the normal equation by QR factorization are:

- QR is the most numerically stable solution [40];
- QR prevents squaring H matrix in the normal equation;
- QR can be implemented with ordering to reduce fill-in (i.e., Tinney Scheme 2).

The price that one has to pay for such a numerically stable algorithm is its computational burden.

Krylov subspace iterative methods are methods of choice for many problems involving large-sparse systems of linear equations. Power systems state estimation is one such example. While there is no guarantee that if proven to be reliable on one sparse problem, an iterative method would be reliable in general, there is hope that in the large set of Krylov subspace methods some of them would perform well on our problem. Although present for many years, Krylov subspace iterative methods did not receive much attention from power system state estimation researchers.

The benefits of iterative methods for the solution of large sparse system are well recognized. Among them, the most prominent are:

- theoretical convergence in a finite number of steps that is (sometimes significantly) smaller than the order of the system;
- the original sparsity pattern is preserved;
- only matrix-vector products are required;
- can be implemented without explicitly knowing the coefficient matrix ("matrix-free").

In general, iterative methods are recommended when direct methods produce excessive fill-in or when the coefficient matrix (i.e., the Jacobian or Hessian) is not explicitly available. While in power system state estimation, the coefficient matrix is available and well defined, the problem of fill-in exists. In large-sparse problems, direct methods tend to increase matrix density and thus incur additional work. The best one can do is to keep fill-in under control by using ordering algorithms.

An ordering algorithm permutes the rows and columns of a matrix so that the number of fill-ins during factorization is minimized. In a seminal paper on the application on Givens rotations to power system state estimation [93], Vempati, Slutsker and Tinney investigated several schemes for column and row ordering of H. Their conclusion was that the best scheme consisted of minimum degree ordering for $H^T R^{-1} H$ to determine column ordering of H followed by *staircase* ordering with a row count tie-breaker rule to order the rows of H. Practical direct solvers are dependent on effective ordering algorithms, while iterative methods preserve initial sparsity.

Our motivation is to extend the pool of iterative methods applied to power system state estimation in the hope of maintaining the state estimator's reliability and speed. This work is intended to screen iterative methods and to assess their performance on the power system state estimation problem. Due to the nature of our problem, we concentrate our search on the methods known to be

successful least squares or normal equation solvers. Prospective Newton-Krylov methods will then be tested against Newton-QR which is numerically the most stable method and performance will be assessed.

Therefore Krylov subspace methods that have been proposed to solve least squares problem will be our target. Ideally, the Krylov subspace method would

- not need to "square" the H matrix and deteriorate conditioning;
- preserve the numerical stability of the direct method (i.e., QR factorization);
- have a well-defined and efficient preconditioner that incurs minimal additional cost;
- be computationally cheaper than the direct method.

Power system state estimation has been traditionally solved by direct methods. The first to apply conjugate gradient methods to power system state estimation were Nieplocha and Carroll in [64]. Their work has shown that, when implemented with proper sparse matrix format, preconditioned conjugate gradients (PCG) are competitive to a direct solver. Further, PCG methods posses properties that can enhance the speed of calculations on parallel processing computers.

Galiana *et al.* in [31] applied the conjugate gradient method with an incomplete Cholesky preconditioner to solve sets of linear equations in the fast decoupled and the DC load flow problems. Their test results show that PCG performs significantly faster than a direct solver as the system size and connectivity increases.

A review of the important aspects of Krylov subspace methods and its fundamental ideas relative to power flow applications is presented by Semlyen in [78].

Dağ and Alvarado in [22] proposed a method for obtaining a positive definite incomplete Cholesky preconditioner for coefficient matrices that arise in power system applications like state estimation, power flow, security analysis, and transient stability. They demonstrate reliable convergence of the CG method with their proposed preconditioner.

Dağ and Samlyen in [23] proposed preconditioned conjugate gradient with an approximate inverse of the coefficient matrix as preconditioner, based on a matrix-valued Chebyshev polynomial. With the proposed PCG method they solved fast decoupled load flow. Their test results showed that the PCG algorithm with matrix-valued Chebyshev polynomial as a preconditioner is comparable to traditional direct methods used for fast decoupled load flow. Their opinion is that if implemented using parallel processing architecture, their proposed algorithm could perform even better.

Nieplocha *et al.* in [65] compared performance of a direct versus a CG-based solver of the state estimator's normal equations on multi-core-processor computers. Their implementation showed encouraging results in favor of the CG solver.

The general view of all of these authors is that problem-specific preconditioners deserve more research because of their promise to improve convergence properties of the CG methods.

3.1.1 Power System State Estimation - Problem Formulation

Power system state estimation is an algorithm for determining the system state from a model of the power system network and redundant system measurements. The state estimation nonlinear measurement model is defined by

$$z = h(x) + \epsilon$$

The state estimation problem is formulated as a weighted least-squares problem

$$\min_{x \in \mathbb{R}^n} J(x) = \frac{1}{2}(z - h(x))^T R^{-1}(z - h(x))$$

The problem is solved by minimization of the quadratic approximation of the objective function around a starting point. The first-order necessary conditions for a minimum result in the equation

$$\nabla J(x) = -H^T R^{-1}(z - h(x)) = 0$$

The optimum is found via Newton's method by solving the system

$$\nabla^2 J(x_k)s = -\nabla J(x_k)$$

$$x_{k+1} = x_k + s$$

at each iteration, until convergence is attained. In practice, the exact Hessian $\nabla^2 J(x)$ is approximated by the Gauss-Newton Hessian $\nabla^2 J(x) = H^T R^{-1} H$, resulting in an iterative equation of the form

$$H^T R^{-1} H s = H^T R^{-1} r \tag{3.1}$$

where $H = \partial h / \partial x \in \mathbb{R}^{m \times n}$ is the Jacobian matrix and $r = z - h(x)$ is the m-dimensional residual vector.

Equations (3.1) are the so-called normal equations of the weighted least-squares problem. While the normal equations can be solved using several methods, orthogonal transformations (i.e., QR

decomposition) is numerically the most stable direct method. QR factorization can be computationally expensive even for sparse problems. One reason for this is the creation of fill-ins during the factorization process (a fill-in is the creation of a new non-zero matrix element). With direct methods, the ability to overcome fill-ins is limited. The best one can do with direct methods is to keep fill-ins under control by ordering algorithms.

Krylov subspace iterative methods are well known solvers for large sparse linear systems. Among the benefits of iterative methods for the solution of large sparse systems are: only matrix-vector multiplications are required per iteration; there are no fill-ins; theoretical convergence within at most n iterations (using exact arithmetic), where n is the size of the system, though in practice they may require far fewer or far more than n iterations. The hope is that the state estimator can take advantage of that. The conjugate gradient method works on symmetric positive-definite systems, such as equation (3.1), although, as in the direct methods, a concern is the squared condition number of H when applied to normal equation.

The use of preconditioners has clearly been the key to the success of CG methods in practice. It has been found in [22] that the preconditioner has to be positive definite to ensure convergence. Having a symmetric and positive definite gain matrix ($H^T R^{-1} H$) is a necessary but not a sufficient condition to obtain a positive definite incomplete Cholesky preconditioner. The LSQR method [68], [67] solves the normal equations without squaring the H matrix.

3.1.2 Sparse matrix computation - The Problem of Fill-in

Power system network equations require the use of large sparse matrices. A matrix is considered sparse if most of its elements are zero. The reasons for development of sparse matrix methods are to reduce storage and computational requirement. Sparse matrix problems require special techniques which avoid or reduce the storage of zero elements and work only with the nonzero entries. A historical review of the sparse matrix methods relative to power system applications is provided by Alvarado *et al.* in [7].

When using matrix factorization in either the dense or sparse case, zero elements before factorization can become nonzero after factorization. The phenomenon of turning a zero element of a sparse matrix into a nonzero element during a factorization is called *fill-in*. This kind of behavior occurs in any kind of factorization (i.e., Cholesky, QR, ...). For full matrices this phenomenon is not critical since all elements are stored in spite of their value. For sparse matrices this is not the case. Fill-ins increase storage requirements and produce an additional computational burden.

The goal of sparse matrix factorization is to limit the fill-in as much as possible. The applied mathematics community has developed algorithms that minimize fill-ins. These algorithms order the rows and columns of a given matrix A with the aim of reducing the fill-in during the factorization, prior to the actual factorization. Thus, the factorization process is divided into two stages: the first is symbolic, and the second is referred to as the numeric stage. Symbolic factorization is applied to the basic sparsity structure of the matrix A without regard for the numerical values of its entries.

3.1.3 Condition Number Analysis

Condition number analysis of the problem equation is important whether direct or iterative methods are used. A poorly conditioned problem is generally difficult to solve by any method.

For a square matrix $A \in \mathbb{R}^{n\times n}$, *the 2-norm condition number* is:

$$\kappa_2(A) = \|A^{-1}\|_2 \cdot \|A\|_2 = \frac{\lambda_1(A)}{\lambda_n(A)}$$

where: $\|.\|_2$ is an Euclidean or 2-norm and $\lambda_1(A) \geq \lambda_2(A) \geq \cdots \geq \lambda_n(A)$ are eigenvalues of A. In general, if $A \in \mathbb{R}^{m\times n}$ is a non-square matrix, with p-singular values $\sigma_1(A) \geq \sigma_2(A) \geq \cdots \geq \sigma_p(A)$, where $p = \min\{m, n\}$ then the *2-norm condition number* of A is defined as:

$$\kappa_2(A) = \frac{\sigma_1(A)}{\sigma_p(A)}$$

The condition number measures the relative change in the solution as a multiple of the relative change in the data. In other words, for the linear system $Ax = b$ relative error in x can be $\kappa_2(A)$ times the relative error in A and b [35]. Matrices with small condition number are said to be *well-conditioned* while matrices with large condition number are *ill-conditioned*.

The convergence rate of Krylov subspace methods depends on condition number; and they perform poorly on systems that are not well conditioned. Besides the conditioning of the problem, the convergence of iterative methods also depends on the spectral properties or the distribution of the eigenvalues of the coefficient matrix [9].

Condition number analysis of power system state estimation has been the subject of research first by Gu *et al.* in [36] and then Ebrahimian and Baldick in [28]. Reference [28] studied the effect of combinations of different types of measurements on the condition number of the gain matrix. In [28] a formula for the approximate condition number in terms of number of different measurement types is developed. It is based on the following assumptions: the state estimator Jacobian is derived from the fast decoupled load flow model, the network is radial, and different measurement error

variances are assigned to different measurement types. These assumptions are similar to the ones in [36]. Reference [28] provides guidelines for the order of the condition number that can be expected in power system state estimation.

With exact arithmetic, CG would terminate in at most n iterations. In practice (finite precision arithmetic) it may need far more, or far fewer if A has clustered eigenvalues.
CG works well on matrices that are either well conditioned or have just a few distinct eigenvalues (eigenvalues or singular values are clustered). A favorable eigenvalue distribution can be achieved by finding a preconditioner, a topic to be discussed in this chapter.

Table 3.1 presents spectral properties and condition numbers for the matrices derived from the IEEE 14-bus and IEEE 30-bus test cases described in Fig. A.1 and in Fig. A.2 of Appendix A, where more details about the particular case can be found. Note that the matrix $H_\omega = R^{-1/2}H$ is the weighted Jacobian matrix used in Newton-QR algorithm, and G is the gain matrix $H^T R^{-1} H$.

Table 3.1: Condition Number and spectral properties of the IEEE test cases

	IEEE 14 bus	**IEEE 30 bus**
$\kappa_2(H)$	77.8	342.3
$\kappa_2(H_w)$	35.3	190.3
$\kappa_2(G)$	$1.15 \cdot 10^3$	$3.16 \cdot 10^4$
$\sigma_1(H)$	50.6	88.9
$\sigma_n(H)$	0.641	0.259
$\sigma_1(H_w)$	$1.56 \cdot 10^3$	$2.8 \cdot 10^3$
$\sigma_n(H_w)$	45.1	14.76
$\lambda_1(G)$	$2.41 \cdot 10^6$	$6.81 \cdot 10^6$
$\lambda_n(G)$	$2.1 \cdot 10^3$	215.6

3.2 Krylov Subspace Methods

Consider the system of linear equations $Ax = b$. The kth Krylov subspace generated by the matrix A and vector r_0 is

$$\mathcal{K}_k(A, r_0) \equiv span\left\{r_0, Ar_0, A^2 r_0, \ldots, A^{k-1} r_0\right\}$$

where r_0 is the initial residual vector $r_0 = b - Ax_0$ associated with the initial approximate solution x_0. A Krylov subspace method determines

$$x_k = x_0 + z_k = A^{-1}b \qquad z_k \in \mathcal{K}_k \qquad \text{for } k \leq n$$

where

$$z_k = \sum_{j=0}^{k-1} \lambda_j A^j r_0 \in \mathcal{K}_k$$

Different methods are determined by different choices of z_k. Krylov subspace methods are based on two traditional criteria:

1. Minimal residual (MR) criteria stated as: choose $z_k \in \mathcal{K}_k$ to solve

$$\min_{z \in \mathcal{K}_k} \|b - A(x_0 + z)\|_2 = \min_{z \in \mathcal{K}_k} \|r_0 - Az\|_2$$

2. Orthogonal residual (OR) criteria stated as: Choose $z_k \in \mathcal{K}_k$ so that

$$\begin{aligned} r(z_k) &= b - A(x_0 + z_k) \perp \mathcal{K}_k \\ &= r_0 - Az_k \perp \mathcal{K}_k \end{aligned}$$

Since the CG method, that we will use in this chapter is based on the OR criterion, we will state the basic idea behind it. For a given basis matrix $B_k = (b_1, \cdots, b_k)$ of the kth Krylov subspace, the vector $z_k \in \mathcal{K}_k$ can be written as $z_k = B_k y_k$ for some $y_k \in \mathbb{R}^k$. With respect to the basis B_k, the OR criterion can be stated as

$$B_k^T A B_k y_k = B_K^T r_0 \tag{3.2}$$

The original Krylov subspace basis $B_k = \left(r_0, Ar_0, \cdots, A^{k-1} r_0\right)$ is often very ill-conditioned. A well-conditioned basis of the Krylov subspace V_k is generated with the Arnoldi process outlined in Alg. 4. The basis generated by the Arnoldi process is orthonormal (i.e., $V_k^T V_k = I$) because it draws on the modified Gram Schmidt algorithm.

The Arnoldi process of Algorithm 4 generates

$$V_k = (v_1, \cdots, v_k) \qquad \text{and} \qquad H_k = \begin{pmatrix} h_{11} & \cdots & h_{1k} \\ h_{21} & & \vdots \\ \vdots & \ddots & \vdots \\ 0 & \cdots & h_{k+1,k} \end{pmatrix}$$

For some k the process breaks down (i.e., $h_{k+1,k} = 0$)

$$Av_k \in \mathcal{K}_k = span\left\{v_1, , \ldots, v_k\right\}$$

$$AV_k = \begin{cases} V_{k+1} H_k & \text{before breakdown} \\ V_k \bar{H}_k & \text{on breakdown} \end{cases}$$

Algorithm 4 Arnoldi process [95]

`Given` r_0

`set` $\rho_0 \equiv \|r_0\|$ `and` $v_1 \equiv \frac{r_0}{\rho_0}$

for $k = 1, 2, \ldots$ **do**

`Initialize` $v_{k+1} = Av_k$

for $i = 1, \ldots, k$ **do**

`Set` $h_{ik} = v_i^T v_{k+1}$

`Update` $v_{k+1} \leftarrow v_{k+1} - h_{ik} v_i$

end for

$h_{k+1,k} = \|v_{k+1}\|_2$

`Update` $v_{k+1} \leftarrow \frac{v_{k+1}}{h_{k+1,k}}$

end for

where

$$\bar{H}_k = \begin{pmatrix} h_{11} & \cdots & \cdots & h_{1k} \\ h_{21} & h_{22} & \cdots & h_{2k} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & h_{k,k-1} & h_{kk} \end{pmatrix}$$

is a $k \times k$ upper Hessenberg matrix. After a Krylov subspace orthonormal basis V_k is found, the OR condition in equation (3.2) becomes

$$V_k^T A V_k y_k = V_k^T r_0$$

which leads to

$$\bar{H}_k y_k = \rho_0 e_1$$

where $\rho_0 = \|r_0\|$ and $e_1 = (1, 0, \cdots, 0)^T$. Therefore we only need to solve a $k \times k$ Hessenberg system. Once y_k is obtained, z_k is found from $z_k = V_k y_k$

$$x_k = x_0 + z_k = A^{-1} b$$

A general reference on Krylov subspace methods is Saad [73].

A *Newton-Krylov method* is an implementation of Newton's method in which a Krylov subspace method is used to approximately solve the linear systems that characterize the steps of Newton's

method. In our case the Krylov subspace method will be applied to solve the normal equation (3.1) of the least squares problem. A Newton-Krylov method that uses a specific Krylov subspace method is often designated by appending the name of the method to "Newton", as in "Newton-CGNR" or "Newton-LSQR".

In this chapter the name Newton-QR is reserved for the direct method of solving the normal equations based on Givens rotations, against which we compare the performance of the Newton-Krylov method.

3.2.1 The Conjugate Gradient Method

The conjugate gradient (CG) method was developed by Hestenes and Stiefel in [38] and is one of the best known iterative methods for symmetric positive definite (SPD) systems of linear equations. The CG method is a realization of the OR criterion that requires

$$r(z_k) \quad = \quad b - A(x_0 + z_k) \perp \mathcal{K}_k$$

CG exploits orthogonality of the Krylov basis to estimate the residuals. In finite-precision arithmetic this orthogonality can be lost and the estimate of the residual in the iteration can be poor [48].

For a symmetric matrix A, a step z_k that satisfies the orthogonal residual criterion can be found from the following constraint minimization problem:

$$z_k = \min_{z \in \mathcal{K}_k} \Phi(z) = \frac{1}{2} (z - x_*)^T A (z - x_*) = e^T A e \tag{3.3}$$

where: $x_* = A^{-1}b - x_0 = A^{-1}r(x_0)$.

In order to show that the solution of this constrained minimization problem satisfies the OR criterion, we define the function $\Psi(y) = \Phi(V_k y)$ where $z_k = V_k y$ and transform the above constrained problem in $\mathcal{K}_k$ into an equivalent unconstrained problem in $\mathbb{R}^k$

$$y_k = \min_{y \in \mathbb{R}^k} \Psi(y) = \min_{y \in \mathbb{R}^k} \Phi(V_k y)$$

The first-order necessary condition for the above problem is

$$\nabla \Psi(y) = V_k^T \nabla \Phi(V_k y) = V_k^T \nabla \Phi(z) = 0$$

or in other words, the gradient of $\Psi(y)$ is orthogonal to the basis of the Krylov subspace. By choosing a proper scalar valued function $\Phi(z)$ with its gradient equal to the residual $r(z)$, the first-order necessary condition provides us with the OR criterion. The objective function in (3.3) is the

desired function since

$$\nabla\Phi(z) = A(z - x_*) = Az - r(x_0) = -r(z)$$

Therefore minimization of $\Psi(y)$ with $z_k = V_k y$, is equivalent to finding z_k that satisfies the OR criterion

$$\begin{aligned} \nabla\Psi(y) = 0 \quad &\Leftrightarrow \quad V_k^T \nabla\Phi(z) = 0 \\ &\Leftrightarrow \quad V_k^T r(z) = 0 \end{aligned}$$

As we will see shortly, the underlying idea behind the LSQR method is very similar, although the solution steps in obtaining z_k are different.

Two important theorems in [20] describe factors that influence the convergence behavior of the CG method. The first factor is the condition number of A. If the matrix is ill-conditioned, then round-off errors may prevent the algorithm from obtaining a sufficiently accurate solution after n steps. The worse the conditioning of A, the slower the convergence of CG is likely to be. The second factor is the eigenvalue distribution of A. The tighter the eigenvalues of A are clustered, the faster the convergence.

The pseudocode for the Conjugate Gradient Method is given in Algorithm 5.

Algorithm 5 The Conjugate Gradient Method [95]

`Given:` A, b, x, tol, $itmax$

`Initialize:` $r = b - Ax$, $\rho^2 = \|r\|_2^2$, $z = 0$, $\beta = 0$

`Iterate`

for $itno = 1, 2, \ldots, itmax$ **do**

`If` $\rho \leq tol$, `go to END`

`Update` $p \leftarrow r + \beta p$

`Compute` Ap

`Compute` $p^T Ap$ `and` $\alpha = \rho^2 / p^T Ap$

`Update` $z \leftarrow z + \alpha p$

`Update` $r \leftarrow r - \alpha Ap$

`Update` $\beta \leftarrow \|r\|_2^2 / \rho^2$ `and` $\rho^2 \leftarrow \|r\|_2^2$

end for

`Update` $x \leftarrow x + z$

3.2.2 The CG for the solution of the Normal Equation

The conjugate gradient method applied to the normal equation

$$A^T Ax = A^T b \tag{3.4}$$

constructs the k^{th} iterate

$$x_k = x_0 + z_k \qquad \text{for } k \leq n$$

using the update z_k that lies in the Krylov subspace

$$\widetilde{\mathcal{K}}_k(A^T A,\ \ A^T r_0) \equiv \text{span}\left\{A^T r_0,\ \ (A^T A)A^T r_0,\ \ \ldots\ \ (A^T A)^{k-1} A^T r_0\right\}$$

When applied to the system $Ax = b$, the CG method produces z_k such that $e^T Ae$ is minimal over all corrections in $\mathcal{K}_k$. In the same way, when applying the CG method to the normal equation (3.4), one has to solve the minimization problem

$$z_k = \min_{z \in \mathcal{K}_k} = \widetilde{e}^T (A^T A)\widetilde{e}$$

where: $\widetilde{e} = z_k - (A^T A)^{-1} A^T b$. It can be shown that this minimization problem is equivalent to the problem

$$z_k = \min_{z \in \mathcal{K}_k} (Az_k - b)^T (Az_k - b) = \min_{z \in \mathcal{K}_k} \|r_k\|^2 \tag{3.5}$$

Therefore the kth iterate minimizes $\|r_k\|^2$ over all corrections in $\widetilde{\mathcal{K}}_k$. Hence the name CGNR, meaning CG on the Normal equations with Residual minimization.

Although sometimes effective, solving the normal equation using the CG method is handicapped by the squaring of the condition number of A [35].

3.2.3 Preconditioning

A *preconditioner* is a matrix M that transforms the initial problem $Ax = b$ into an equivalent system

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

whose coefficient matrix has more favorable spectral properties. The preconditioner clusters the eigenvalues and improves the condition number, which ultimately speeds the convergence of the equivalent system. Iteration of a preconditioned system is more expensive, but with careful choice

of preconditioner, one may reduce the total number of iterations. Finding an efficient preconditioner is a difficult task.

While there are preconditioners that are inexpensive to construction, more often than not the use of a preconditioner in an algorithm involves the extra cost in finding and perhaps factoring M. Many times the only effective preconditioner M is one that approximates A. In those cases, in order to avoid the need for factoring, a preconditioner is defined as $M = LU$ (with L and U as triangular matrices). That way solving a preconditioned system would be comparable in expense to solving a system with coefficient matrix A. In preconditioned algorithms, the hope is that the initial cost in obtaining the preconditioner will pay off through the iterative process. Another computational savings might be repeated use of the same preconditioner in successive iteration steps. An effective preconditioner is often problem specific and thus difficult to find. Each iteration of the preconditioned CG method in a dense system requires:

- one precondition solve via forward/backward substitution ($\mathcal{O}\ (n^2)$)
- one Av product ($\mathcal{O}\ (n^2)$)

The Incomplete Factorization Preconditioner

A very important and widely used class of preconditioners is based on incomplete Cholesky factorization of the coefficient matrix. During Cholesky factorization certain fill elements are created. Incomplete Cholesky strategies range from discarding any fill-in during the sparse Cholesky factorization to allowing different levels of fill-in.

While incomplete Cholesky is not too expensive, one has to be very careful with its implementation. Obtaining a stable algorithm for incomplete Cholesky is a non-trivial task. One may assume that it is easy to provide incomplete Cholesky just by constraining the level of fill the existing algorithm but it has been shown by Golub and Van Loan in [35] that such an algorithm may not be stable. Incomplete Cholesky may encounter division by zero pivot or may result in an indefinite matrix. (matrix M is indefinite if $x^T Mx < 0\ \forall x \neq 0$). For a stable algorithm we refer the reader to Elman [29].

In order to be efficient, an incomplete Cholesky preconditioner must be positive definite ($x^T Mx > 0\ \forall x \neq 0$). As stressed by Dağ and Alvarado in [22], obtaining a positive definite incomplete Cholesky preconditioner is a nontrivial task even for a matrix that is symmetric positive definite.

The CGNR simulation results

The CGNR method is employed for solving the normal equation

$$H^T R^{-1} H s = H^T R^{-1} r$$

of the state estimation problem.

The following example will illustrate the performance of the CGNR method on the power system state estimation problem. The CGNR method has been tested on the IEEE 14- and 30-bus networks with measurement sets described in Fig. A.1 on page 134 and in Fig. A.2 on page 136, respectively, of Appendix A. For performance comparison, the Newton-QR direct method has been used. Using the Krylov subspace methods with exact arithmetic, the number of iteration is constrained by the size of the system to be solved. In our case the size of the system is determined by the number of state variables, which is $2n-1$ where n is the number of buses in the network. In practice we may expect that the number of iterations per step to be larger or smaller than n.

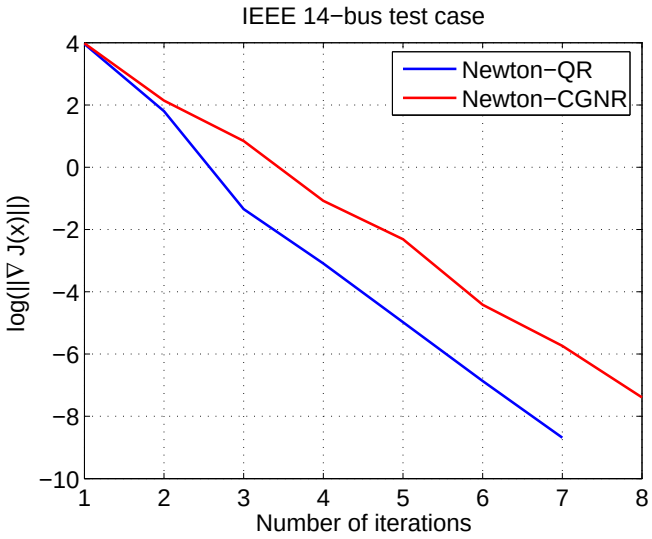

Figure 3.1: Convergence performance of the CGNR method for the IEEE 14-bus test case

The first test case is the IEEE-14 bus network with the measurement set shown in Fig. A.1 on page 134 in Appendix A. The size of the system is $2n-1=27$, which implies that in exact arithmetic CGNR would converge within 27 iterations. Convergence of the CGNR method is shown in Fig. 3.1. The statistics showing the CGNR work per iteration are presented in Table 3.2. The CGNR results have shown that residual reduction is obtained many times for a number of inner

Table 3.2: Newton-CGNR applied on IEEE 14-bus test case

Iteration	# of inner iterations
1	18
2	17
3	23
4	16
5	27
6	13
7	30
8	18

iterations less than 27, although in outer iteration 7, the number of inner iterations needed is slightly greater than 27.

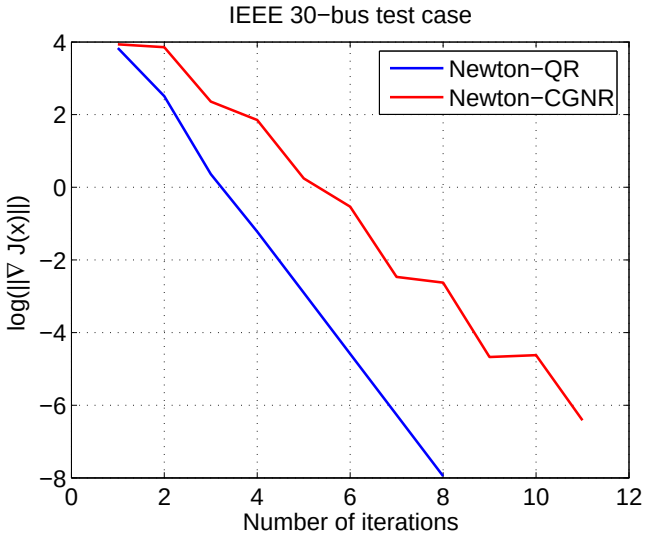

Figure 3.2: Convergence performance of the CGNR method for IEEE 30-bus test case

For the IEEE 30-bus test case and the measurement set shown in Fig. A.2 page 136 of Appendix A, results are shown in Table 3.3 and in Fig. 3.2. In this case, although successful, the number of iterations often exceeds by far the size of the system (i.e., $2n - 1 = 59$).

3.3 The LSQR Method

The LSQR method, which is similar in style to the CG method applied to the normal equation, will be discussed in this section. The LSQR algorithm has been proposed by Saunders and Paige

Table 3.3: Newton-CGNR applied on IEEE 30-bus test case

Iteration	**# of inner iterations**
1	48
2	68
3	48
4	82
5	61
6	91
7	47
8	117
9	47
10	114
11	47

in [68] and [67]. Numerical tests that compare LSQR with several other CG algorithms are given in [68]. It is shown that LSQR is more reliable than any other CG based method when A is ill-conditioned.

As mentioned before, conjugate gradients (CG) work on symmetric positive-definite systems. When applied to the normal equation, the concern is the squared condition number. It would be advantageous if a Krylov subspace iterative method could solve the least-squares normal equation without squaring the A matrix. The reasoning is just as with the QR factorization direct method: the most effective and robust methods for solution of least-squares problem prevent squaring the gain matrix and work with A directly.

LSQR is a method that meets this requirement. It is a Krylov subspace iterative method, analytically equivalent to the standard method of conjugate gradients. LSQR solves the normal equation without squaring the gain matrix, and so it possesses more favorable numerical properties. The difference between CG for least-squares and LSQR is that CG is based on the Lanczos method while LSQR is based on the Golub-Kahan bidiagonalization process. The matrix operations needed to perform the LSQR algorithm are products: Av and $A^T u$, k-Givens rotations, and forward substitution. The computational expense associated with the algorithm is of the order n^2 in a dense case.

First we will discuss the LSQR method applied to the system

$$Ax = b$$

where A is an $m \times n$ real matrix so that $m \geq n$ and b is a real $m \times 1$ vector. A rectangular $m \times n$

matrix A can be reduced to lower bidiagonal form by

$$U^T AV = B$$

where U $(m \times k)$ and V $(n \times k)$ are orthogonal matrices and B is a $k \times k$ lower bidiagonal matrix. An algorithm that brings A into bidiagonal form is known as the Golub-Kahan process [34]. The method will be first described theoretically and then implementation details will be addressed.

3.3.1 Golub-Kahan bidiagonalization process

Bidiagonalization is often used as the first step for dense singular value decomposition (SVD) [35]. The approach to bidiagonalizing A is to generate columns of U and V sequentially as is done by the Lanczos algorithm for tridiagonalizing a symmetric matrix used in the CG algorithm. So in a sense the Golub-Kahan bidiaginalization algorithm is a Lanczos-type algorithm. The algorithm for the Golub-Kahan process generates vectors u_k and v_k and positive scalars α_k and β_k $(k = 1, 2, \dots)$ as described by the following equations and outlined in Algorithm 6. Bidiagonalization is performed iteratively and requires products Av and $A^T u$; therefore, sparsity can be fully utilized. The scalars

Algorithm 6 Golub-Kahan process

set $\beta_1 = \|b\|$ and $u_1 = \frac{b}{\beta_1}$ (exit if $\beta_1 = 0$)

set $\alpha_1 = \|A^T u_1\|$ and $v_1 = \frac{A^T u_1}{\alpha_1}$ (exit if $\alpha_1 = 0$)

Iterate

for $k = 1, 2, \dots$ **do**

$u_{k+1} = Av_k - \alpha_k u_k$

$\beta_{k+1} = \|u_{k+1}\|$

$u_{k+1} \leftarrow \frac{1}{\beta_{k+1}} u_{k+1}$

Exit when $\beta_{k+1} = 0$

$v_{k+1} = A^T u_{k+1} - \beta_{k+1} v_k$

$\alpha_{k+1} = \|v_{k+1}\|$

$v_{k+1} \leftarrow \frac{1}{\alpha_{k+1}} v_{k+1}$

Exit when $\alpha_{k+1} = 0$

end for

$\alpha_i \geq 0$ and $\beta_i \geq 0$ are chosen so that $\|u_i\| = |v_i\| = 1$. The algorithm recurrence relationships can

be written also as

$$\left.\begin{array}{l} u_{k+1} = Av_k - \alpha_k u_k \\ \beta_{k+1} = \|u_{k+1}\| \\ u_{k+1} \leftarrow \frac{1}{\beta_{k+1}} u_{k+1} \end{array}\right\} \quad \Leftrightarrow \quad \beta_{k+1} u_{k+1} = Av_k - \alpha_k u_k \qquad \text{for} \quad k = 1, 2, \ldots$$

and

$$\left.\begin{array}{l} v_{k+1} = A^T u_{k+1} - \beta_{k+1} v_k \\ \alpha_{k+1} = \|v_{k+1}\| \\ v_{k+1} \leftarrow \frac{1}{\alpha_{k+1}} v_{k+1} \end{array}\right\} \quad \Leftrightarrow \quad \alpha_{k+1} v_{k+1} = A^T u_{k+1} - \beta_{k+1} v_k$$

Therefore, after k steps we have:

$$AV_k = U_{k+1} B_k = U_k L_k + \beta_{k+1} u_{k+1} e_k^T$$

$$A^T U_{k+1} = V_k B_k^T + \alpha_{k+1} v_{k+1} e_{k+1}^T = V_{k+1} L_{k+1}^T$$

where $U_k = (u_1 \;\; u_2 \;\; \ldots \;\; u_k)$, $V_k = (v_1 \;\; v_2 \;\; \ldots \;\; v_k)$, L_k is lower bidiagonal, and B_k is lower bidiagonal with one extra row:

$$B_k = \begin{pmatrix} \alpha_1 & & & & \\ \beta_2 & \alpha_2 & & & \\ & \beta_3 & \alpha_3 & & \\ & & \ddots & \ddots & \\ & & & \beta_k & \alpha_k \\ & & & & \beta_{k+1} \end{pmatrix} \qquad L_k = \begin{pmatrix} \alpha_1 & & & & \\ \beta_2 & \alpha_2 & & & \\ & \beta_3 & \alpha_3 & & \\ & & \ddots & \ddots & \\ & & & \beta_k & \alpha_k \end{pmatrix}$$

thus, B_k can be written as

$$B_k = \begin{pmatrix} L_k \\ \beta_{k+1} e_k^T \end{pmatrix}$$

The bidiagonalization process breaks down for some $k \leq n$ (i.e. either $\beta_{k+1} = 0$ or $\alpha_{k+1} = 0$). Using exact arithmetic $U_k^T U_k = I$ and $V_k^T V_k = I$, while in the presence of rounding errors, the previous identities hold to within machine precision. In practice using floating-point calculations, more sophisticated stopping criterion is needed since β_{k+1} is unlikely to vanish for any k.

$$AV_k = \begin{cases} U_{k+1} B_k & \text{before breakdown} \\ U_k L_k & \text{on breakdown} \end{cases}$$

3.3.2 The LSQR Algorithm

Development of the LSQR algorithm starts with the same objective function as the CGNR method. The objective function is least-squares residual minimization over the vectors in the kth Krylov subspace

$$J(x) = \min_{x_k \in \mathcal{K}_k} r_k^T r_k = \min_{x_k \in \mathcal{K}_k} \|r_k\|^2$$

where

$$r_k = b - Ax_k$$

is the residual vector for a given x_k. If we choose $y_k \in \mathbb{R}^k$ such that $x_k = V_k y_k$ and V_k is a basis of the Krylov subspace $\mathcal{K}_k$, the residual vector can be written as:

$$\begin{aligned} r_k &= b - Ax_k \\ &= \beta_1 u_1 - AV_k y_k \\ &= \beta_1 U_{k+1} e_1 - U_{k+1} B_k y_k \\ &= U_{k+1} (\beta_1 e_1 - B_k y_k) \end{aligned}$$

Thus the unconstrained objective function has the form

$$J(y) = \min_{y \in \mathbb{R}^k} \|U_{k+1} (\beta_1 e_1 - B_k y_k)\|^2$$

Since

$$\|U_{k+1} \cdot w\| = \sqrt{w^T U_{k+1}^T U_{k+1} w} = \|w\|$$

due to orthonormality of the matrix U_{k+1}, the objective function simplifies to

$$J(y) = \min_{y \in \mathbb{R}^k} \|B_k y_k - \beta_1 e_1\|^2 \tag{3.6}$$

with corresponding normal equation

$$B_k^T B_k y_k = B_k^T \beta_1 e_1 \tag{3.7}$$

We only need to solve a $k \times k$ bidiagonal system. Equation (3.7) can be solved using QR factorization of B_k to retain stability. We will use Givens rotations to factor

$$\underbrace{Q_{k,k+1} \ldots Q_{2,3} Q_{1,2}}_{Q_k} B_k = \begin{pmatrix} R_k \\ 0 \end{pmatrix}$$

Note that the QR factorization for LSQR can be computed at negligible cost using k rotations of the lower diagonal elements $\beta_2 \dots \beta_k$ of B_k.

$$\begin{pmatrix} R_k \\ 0 \end{pmatrix} y_k = Q_k^T \beta_1 e_1$$

where

$$Q_k^T \beta_1 e_1 = \begin{pmatrix} z_k \\ \bar{\zeta}_{k+1} \end{pmatrix}$$

so

$$R_k y_k = z_k$$

Substituting back x_k

$$R_k V_k^{-1} x_k = z_k$$

x_k can be calculated by

$$x_k = \underbrace{V_k R_k^{-1}}_{W_k} z_k$$

where

$$W_k = V_k R_k^{-1}$$

can be calculated from

$$R_k^T W_k^T = V_k^T$$

using column-by-column forward substitution. Note that the solution $x_k = V_k y_k$ lies in the Krylov subspace

$$\hat{\mathcal{K}}_k(A^T A, \ A^T b) \equiv \text{span}\left\{ A^T b, \ (A^T A) A^T b, \ \dots \ (A^T A)^{k-1} A^T b \right\}$$

The LSQR algorithm that we implemented is from [68].

Algorithm 7 LSQR algorithm 1

`Initialize` $\beta_1 = \|b\|$ `and` $u_1 = \frac{b}{\beta_1}$

`Initialize` $\alpha_1 = \|A^T u_1\|$ `and` $v_1 = \frac{A^T u_1}{\alpha_1}$

`Iterate`

for $k = 1, 2, \ldots$ **do**

`Bidiagonalization`

$\beta_{k+1} u_{k+1} = A v_k - \alpha_k u_k$

$\alpha_{k+1} v_{k+1} = A^T u_{i+1} - \beta_{k+1} v_k$

`Exit when` $\beta_{k+1} = 0$ `or` $\alpha_{k+1} = 0$

end for

`Factor` $B_k = Q_k R_k$

`Calculate` $z_k = Q_k^T \beta_1 e_1$

`Calculate` W_k `from` $R_k^T W_k^T = V_k^T$

`Calculate` $x_k = W_k z_k$

Table 3.4: LSQR method results for IEEE 14-bus test case

	# of inner iterations for Newton-LSQR with		
outer iteration	**FC precond.**	**IC precond.**	**w/o precond.**
1	1	18	27
2	11	18	27
3	9	15	27
4	6	11	27
5	2	4	18
6		2	7
7			7

Table 3.5: IEEE 14-bus test case - First-order necessary condition

	$\|\nabla J(x)\|$	
iteration	**Newton-LSQR**	**Newton-QR**
1	$1.0196 \cdot 10^{-4}$	$1.0196 \cdot 10^{-4}$
2	94.2197	94.2096
3	0.2578	0.2578
4	$4.299 \cdot 10^{-4}$	$3.2489 \cdot 10^{-4}$
5	$1.8805 \cdot 10^{-4}$	$6.5315 \cdot 10^{-6}$
6	$4.2301 \cdot 10^{-5}$	$6.5474 \cdot 10^{-8}$

Algorithm 8 LSQR algorithm 2[68]

Initialize $\beta_1 = \|b\|$ and $u_1 = \frac{b}{\beta_1}$

Initialize $\alpha_1 = \|A^T u_1\|$ and $v_1 = \frac{A^T u_1}{\alpha_1}$

Set $\omega_1 = v_1$

Set $x_0 = 0$

Set $\overline{\phi}_1 = \beta_1$

Set $\overline{\rho}_1 = \alpha_1$

Iterate

for $i = 1, 2, \ldots$ **do**

Bidiagonalization

$\beta_{i+1} u_{i+1} = A v_i - \alpha_i u_i$

$\alpha_{i+1} v_{i+1} = A^T u_{i+1} - \beta_{i+1} v_i$

Orthogonal transformation

$\rho_i = (\overline{\rho}_i^2 + \beta_{i+1}^2)^{1/2}$

$c_i = \overline{\rho}_i / \rho_i$

$s_i = \beta_{i+1} / \rho_i$

$\theta_{i+1} = s_i \alpha_{i+1}$

$\overline{\rho}_{i+1} = -c_i \alpha_{i+1}$

$\phi_1 = c_i \overline{\phi}_i$

$\overline{\rho}_{i+1} = s_i \overline{\rho}_i$

Update x, ω

$x_i = x_{i-1} + (\phi_i / \rho_i) \omega_i$

$\omega_{i+1} = v_{i+1} - (\theta_{i+1} / \rho_i) \omega_i$

Test for convergence, exit if stopping criteria have been met

end for

3.3.3 The LSQR Simulation Results

To evaluate the performance of the LSQR algorithm we have used the IEEE 14-bus and IEE 30-bus test cases with the measurement sets described in Fig. A.1 on page 134 and in Fig. A.2 page 136 respectively of Appendix A. Preconditioners used in the LSQR algorithm were found once at the beginning of the algorithm and were used repeatedly in successive Newton iteration iterations. Two extreme cases of preconditioners were applied: Full Cholesky (FC) (i.e., option 'inf' in MATLAB meaning "infinite tolerance") and the no fill-ins Cholesky preconditioner or Incomplete Cholesky (IC) (i.e., option '0' in MATLAB, meaning zero tolerance). The FC essentialy requires the full factorization of A, and in terms of computational effort for obtaining and solving the preconditioned system, probably least effective. The FC initially leads to the solution using LSQR in a single iteration ($M^{-1}A = I$); thus the method is the direct method at the very first iteration. Since the preconditioner is calculated only once, consecutive iterations require more work. The Newton-LSQR method without preconditioner was also considered. The incomplete Cholesky (IC) in our case presents a good trade-off between comutational effort for obtaining and solving on one hand and convergence efficiency on the other. Fig. 3.3 illustrates the convergence behavior of the Newton-LSQR method when applied to the IEEE 14-bus test case. Performance of the Newton-LSQR method with Incomplete Cholesky as a preconditioner is depicted in Fig. 3.4. Table 3.5 as well as Fig. 3.4 shows that in the first three iterations of the Newton-QR and the Newton-LSQR are the same.

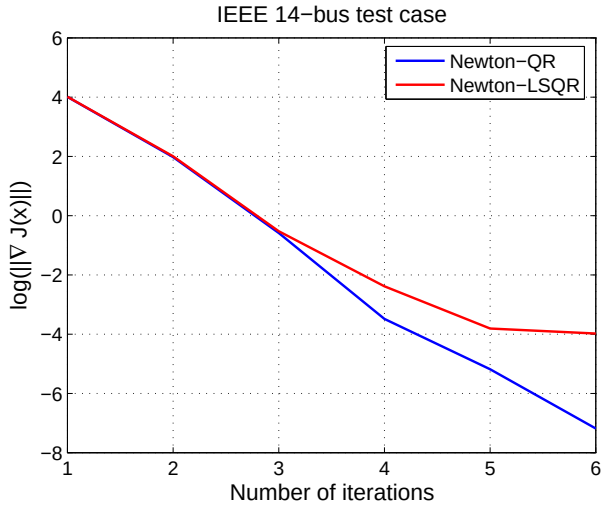

Figure 3.3: Convergence comparison: Newton-QR vs Newton-LSQR for the IEEE 14-bus test case

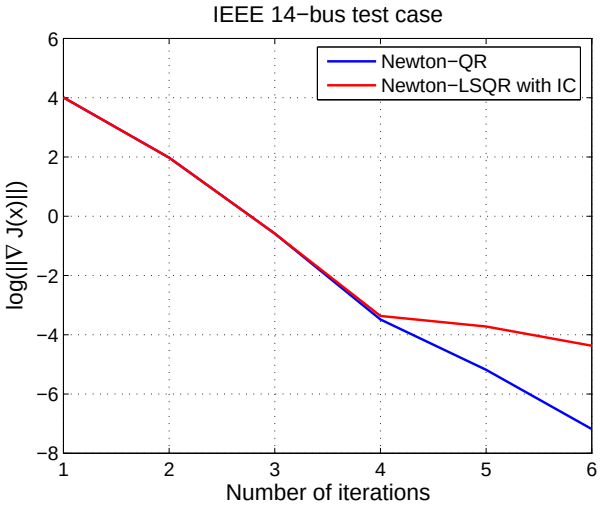

Figure 3.4: Convergence comparison: Newton-QR vs Newton-LSQR with IC preconditioner for the IEEE 14-bus test case

The effect of the IC preconditioner is apparent when comparing the speed of convergence of the test run presented in Fig. 3.5 and the one in Fig. 3.6

Test results for the IEEE 30-bus network depicted in Fig. 3.6 also reveal that LSQR iteration turned out to be exactly equal to the Newton-QR iteration in the first few steps.

An algorithm efficiency comparison has many aspects. Among things to consider are fill-ins, storage reguirements, number of floating point operations, and potential for paralel computer implementation. If proven reliable, the only critical comparison will be floating point work per iteration, since in all other aspects LSQR is far more efficient.

For the sparse case it is more difficult to provide exact floating point operations estimates for either Newton-QR and Newton LSQR. Thus, without loss of generality, we discuss results obtained in terms of the cost of the dense case agorithm. Each LSQR iteration requires $\mathcal{O}\ (n^2)$ arithmetic operation and each QR factorization requires $\mathcal{O}\ (n^3)$ arithmentic operations. Notice that whenever the number of LSQR iterations is less than n for the given outer iteration, the LSQR method cost less. One can see from Table 3.4, that using LSQR with IC preconditioner, number of LSQR iterations per outer iteration is less than n. Overall computational effort in the IEEE 14-bus case for the Newton-LSQR method is $\approx 2.5n^3$ whereas computational effort for the Newton-QR is $5n^3$. Similar conclusions holds for the IEEE 30-bus case: LSQR costs less.

Table 3.6: LSQR method results for the IEEE 30 bus network

	# of inner iterations for Newton-LSQR with		
outer iteration	**FC precond.**	**IC precond.**	**w/o precond.**
1	1	48	59
2	13	48	59
3	11	46	59
4	8	42	59
5	6	28	59
6	3	8	59
7		2	59
⋮			⋮
12			44
13			32
14			13
15			10
16			6
17			3
18			3

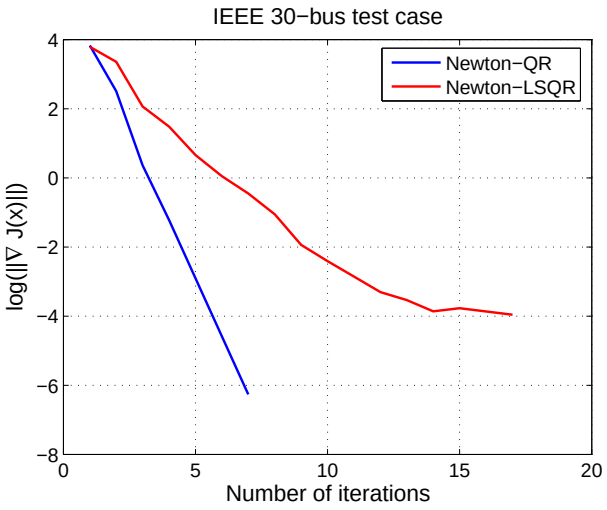

Figure 3.5: Convergence comparison: Newton-QR vs Newton-LSQR for the IEEE 30-bus test case

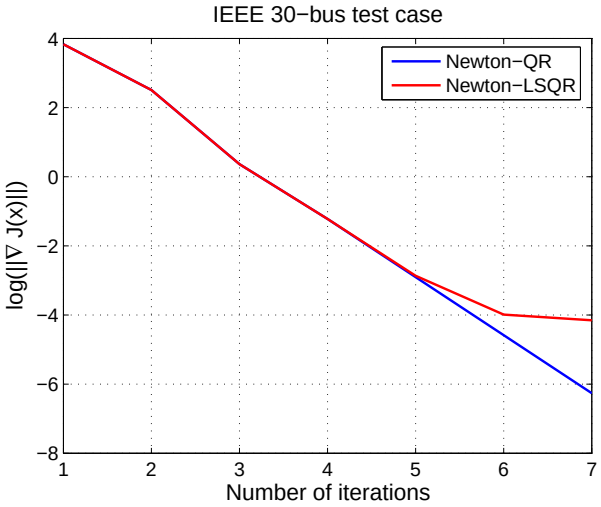

Figure 3.6: Convergence comparison: Newton-QR vs Newton-LSQR with IC preconditioner for the IEEE 30-bus network test case

3.3.4 Conclusions

Preliminary results from the Newton-LSQR iterative solver are very encouraging and interesting. Computational results have shown that the Newton-LSQR method is effective on our test cases with very reasonable computational effort. One may have noticed another asset besides its speed that Newton-LSQR posesses - its numerical robustness. Testing has shown that the reliability of the direct Newton-QR is preserved when an iterative Newton-LSQR method is applied to the state estimation normal equations. The hope remains that this trend will be preserved once Newton-LSQR is applied to larger networks.

Chapter 4

The Use of Importance Sampling in Stochastic OPF

4.1 Introduction

The study of steady-state contingency problems is an important and well recognized activity in the power system planning and operating environment. Methods based upon the use of distribution factors (both active [97] and reactive [45]) are fast and are widely used for studying single-line outages. A well known and computationally efficient technique for contingency ranking is the performance index algorithm [56]. Usually, the quadratic performance index (PI) is a scalar function of either real power loading or voltage magnitude or both. None of these methods go beyond single-line contingencies.

Our intention is to assess multiple credible contingencies while preserving the detailed AC network model of the contingency-constrained OPF (CCOPF). With this model much information can be obtained such as feasibility, locational prices and operating cost. The proposed algorithm combines sequential-quadratic programming for solving CCOPF with a technique called importance sampling [46], [30] for stochastic cost assessment of the multiple contingencies. This method emerged from Monte Carlo importance sampling.

The AC contingency analysis approach is computationally involved, and the computational burden is proportional to the number of contingencies considered. Our approach was to stochastically assess different multiple-contingency scenarios (but not explicitly solve all of them).

Numerous methods have been proposed to find reliable algorithms for contingency selection

and assessment. The key point in any proposed method is to achieve balance between acceptable accuracy and computational speed. The computational burden resulting from contingency analysis is the reason why most studies are limited to single and a few double-line outages. The contribution of this work is in stochastic assessment of multiple contingencies which will allow better modeling of unexpected system events. Currently, reliability is rarely guaranteed under the event of a second contingency. Therefore the cost/pricing aspect of multiple contingency studies is very important in order to determine the cost of reliability. Solutions obtained from these scenarios can be also used to develop appropriate hedging strategies.

Problems associated with contingencies have recently received greater attention, due in part to blackouts around the world. Since low probability scenarios can lead to blackouts and network collapse under certain circumstances, our intention was to go beyond the "$n-1$" criterion. The approach is general and allows the stochastic study of any type of "$n-k$" contingency. Generally speaking, single-line outages are more probable then double or multiple outages. But if the first outage is one of the critical lines, then subsequent outages are more likely. That observation guided our application of the importance sampling algorithm.

This paper presents proposed computational steps for a CCOPF algorithm which is outlined in [51] as well as an importance sampling method [46] for assessment of multiple contingencies. Monte Carlo simulation with importance sampling combined with CCOPF in large networks promises to be an effective technique for analyzing such problems.

4.1.1 Nonlinear CCOPF formulation

The mathematical framework for the solution of the nonlinear contingency constrained optimal power flow (OPF) is based on sequential quadratic programming as proposed in [51] and described in detail in [69]. The contingency constrained optimal power flow minimizes the total cost of a base case operating state as well as the expected cost of recovery from contingencies such as line or generation outages. The sequential quadratic programming (SQP) OPF formulation [63] has been expanded in order to recognize contingency conditions, and the problem is solved as a single entity by an efficient interior point method. The objective function in (4.1) includes the total cost of operation in the pre-contingency or base case as well as the expected cost of recovery from all contingencies. This formulation takes into account the system's corrective capabilities in response to contingencies introduced through ramp-rate constraints.

Contingency constrained OPF is a very challenging problem, because each contingency con-

sidered introduces a new problem as large as the base case problem. Not all contingencies have the same likelihood of occurrence, which leads us to assigning a probability to each contingency considered. The expected cost of these contingencies is defined as

$$E\left\{c_\omega(u_\omega)\right\} = \sum_{\omega=1}^{k} p_\omega d_\omega^T u_\omega$$

Thus, by modeling contingency probabilities we can formulate the optimal power flow as a stochastic programming problem. This formulation is also called the stochastic OPF and its linear form was the subject of the research by Kimball, Clements and Davis in papers [49] and [50] where it was solved via an interior point method and Bender's decomposition.

By proper system reduction and use of constraint relaxation (active set) methods, the computational burden can be reduced significantly. The mathematical formulation of contingency constrained OPF with corrective rescheduling is as follows:

$$\begin{aligned} \text{Minimize} \quad & c(x_0, u_0) + E\left\{c_\omega(u_\omega)\right\} \\ \text{Subject to:} \quad & g(x_0, u_0) = 0 \\ & f(x_0, u_0) \leq 0 \\ & g_\omega(x_\omega, u_\omega) = 0 \\ & f_\omega(x_\omega, u_\omega) \leq 0 \\ & h(u_0, u_\omega) \leq 0 \\ & \omega = 1, \ldots, K \end{aligned} \tag{4.1}$$

where x_0 and x_ω are state variables for the base and contingency cases, respectively, and u_0 and u_ω are pre- and post-contingency control settings. Constraints are following:

$g(x_0, u_0)$	power balance equations for base case;
$f(x_0, u_0)$	set of inequality constraints for base case;
$g_\omega(x_\omega, u_\omega)$	power balance equations for each contingency case;
$f_\omega(x_\omega, u_\omega)$	set of inequality constraints for each contingency case;
$h(u_0, u_\omega)$	ramp-rate constraints;
p_ω	probability of contingency ω;
ω	is the set of possible contingencies $\omega = 1, \ldots K$.

Sequential quadratic programming coupled with an interior point method, as proposed in [63], can be used to solve this optimization problem. The Lagrangian for the above problem with non-negativity constraints imposed on the slack variables s_i and σ_i through a barrier parameter μ,

is

$$\begin{aligned}\mathcal{L} &= c(x_0,u_0) - \mu\left(\sum_{i=1}^{n_{c_0}} \ln s_{0_i} + \sum_{\omega=1}^{K}\sum_{i=1}^{n_{c_\omega}} \ln s_{\omega_i} + \sum_{i=1}^{n_r} \ln \sigma_i\right) \\ &\quad +\lambda_0^T g(x_0,u_0) \\ &\quad +\pi_0^T\left(f(x_0,u_0)+s_0\right) \\ &\quad +\sum_{\omega=1}^{K} \lambda_\omega^T\left(g_\omega(x_\omega,u_\omega)+\pi_\omega^T(f_\omega(x_\omega,u_\omega)+s_\omega)\right) \\ &\quad +\sum_{\omega=1}^{K}(p_\omega d_\omega^T u_\omega + \gamma^T(h(u_0,u_\omega)+\sigma_\omega))\end{aligned} \tag{4.2}$$

A stationary point of the Lagrangian function is a zero of the following system of KKT conditions from the interior point formulation:

$$\begin{aligned}\nabla_{x_0}\mathcal{L} &= \nabla_{x_0}c(x_0,u_0) + G_{x_0}^T\lambda + F_{x_0}^T\pi = 0 \\ \nabla_{u_0}\mathcal{L} &= \nabla_{u_0}c(x_0,u_0) + G_{u_0}^T\lambda + F_{u_0}^T\pi + H_{u_0}^T\gamma = 0 \\ \nabla_{x_\omega}\mathcal{L} &= G_{x_\omega}^T\lambda_\omega + F_{x_\omega}^T\pi_\omega = 0 \\ \nabla_{u_\omega}\mathcal{L} &= G_{u_\omega}^T\lambda + F_{u_\omega}^T\pi + H_{u_\omega}^T\gamma + p_k d_k = 0 \\ \nabla_{\lambda}\mathcal{L} &= g(x_0,u_0) = 0 \\ \nabla_{\pi_0}\mathcal{L} &= f(x_0,u_0) + s_0 = 0 \\ \nabla_{\lambda_\omega}\mathcal{L} &= g_\omega(x_\omega,u_\omega) = 0 \\ \nabla_{\pi_\omega}\mathcal{L} &= f(x_\omega,u_\omega) + s_\omega = 0 \\ \nabla_{\gamma}\mathcal{L} &= h(u_0,u_\omega) + \sigma = 0 \\ \nabla_{s_0}\mathcal{L} &= \pi_0 - \mu S_0^{-1}e = 0 \\ \nabla_{s_\omega}\mathcal{L} &= \pi_\omega - \mu S_\omega^{-1}e = 0 \\ \nabla_{\sigma}\mathcal{L} &= \gamma - \mu\Sigma^{-1}e = 0 \\ & s_0 \geq 0, \quad s_\omega \geq 0, \quad \sigma \geq 0 \\ & \omega = 1,\ldots,K\end{aligned}$$

where $S = \mathrm{diag}(s)$, $S_\omega = \mathrm{diag}(s_\omega)$ and $\Sigma = \mathrm{diag}(\sigma)$ and e is a vector of ones of appropriate dimension. The last three of KKT equations are known as complementary slackness conditions. In order to solve this system of nonlinear equations we first apply a Newton linearization by expanding the KKT equations about x_0, u_0, x_ω, u_ω.

$$
\begin{aligned}
W_{xx}\Delta x + W_{xu}\Delta u + G_x^T\lambda + F_x^T\pi &= -\nabla_x c(x,u) \\
W_{ux}\Delta x + W_{uu}\Delta u + G_u^T\lambda + F_u^T\pi + H_u^T\gamma &= -\nabla_u c(x,u) \\
W_{x_\omega x_\omega}\Delta x_\omega + W_{x_\omega u_\omega}\Delta u_\omega + G_{x_\omega}^T\lambda_\omega + F_{x_\omega}^T\pi_\omega &= 0 \\
W_{u_\omega x_\omega}\Delta x_\omega + W_{u_\omega u_\omega}\Delta u_\omega + G_{u_\omega}^T\lambda_\omega + F_{u_\omega}^T\pi_\omega + H_{u_\omega}^T\gamma &= -p_\omega d_\omega \\
G_x\Delta x + G_u\Delta u &= -g(x,u) \\
F_x\Delta x + F_u\Delta u + s &= -f(x,u) \\
G_{x_\omega}\Delta x_\omega + G_{u_\omega}\Delta u_\omega &= -g_\omega(x_\omega,u_\omega) \\
F_{x_\omega}\Delta x_\omega + F_{u_\omega}\Delta u_\omega + s_\omega &= -f_\omega(x_\omega,u_\omega) \\
\Pi S e &= \mu e \\
\Pi_\omega S_\omega e &= \mu e \\
\Gamma\Sigma e &= \mu e
\end{aligned}
$$

This linearized set of KKT conditions can be seen as necessary conditions of a quadratic optimization problem at each iteration, hence the name sequential quadratic programming.

At this point we give a summary of the major steps of the algorithm. We refer the interested reader to [69] where the complete procedure can be found. The solution procedure is to decompose the system and solve it in a few stages. First we eliminate Δx and λ as well as Δx_ω and λ_ω since these vectors are largest in size. The reduced-order system obtained after elimination of the variables has the form:

$$
\begin{aligned}
\overline{W}_{uu}\Delta u + \overline{F}_u^T\pi + H_u^T\gamma &= \bar{b}_u \\
\overline{W}_{u_\omega u_\omega}\Delta u_\omega + \overline{F}_{u_\omega}^T\Delta\pi_\omega + H_{u_\omega}^T\gamma &= \bar{b}_{u_\omega} \\
\overline{F}_u\Delta u + s &= \bar{b}_\pi \\
\overline{F}_{u_\omega}\Delta u_\omega + s_\omega &= \bar{b}_{\pi_\omega} \\
H_u\Delta u + H_{u_\omega}\Delta u_\omega + \sigma &= b_\gamma \\
\Pi S e &= \mu e \\
\Pi_\omega S_\omega e &= \mu e \\
\Gamma\Sigma e &= \mu e
\end{aligned}
$$

This is still a nonlinear system of equations in terms of s, s_ω, π, π_ω and σ. The next step in the

algorithm is to linearize the system about those variables. Linearized variables Δs, Δs_ω and $\Delta\sigma$ are expressed using the linearized complementary slackness equations and substituted in the rest of the system. After performing that operation and a few algebraic steps, the reduced system will have the following matrix form:

$$\begin{pmatrix} \overline{W}_{uu} & \overline{F}_u^T & 0 & 0 & \cdots & H_u^T \\ \overline{F}_u & -\Pi^{-1}S & 0 & 0 & \cdots & 0 \\ 0 & 0 & \overline{W}_{u_\omega u_\omega} & \overline{F}_{u_\omega}^T & \cdots & H_{u_\omega}^T \\ 0 & 0 & \overline{F}_{u_\omega} & -\Pi_\omega^{-1}S_\omega & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ H_u & 0 & H_{u_\omega} & 0 & \cdots & -\Gamma^{-1}\Sigma \end{pmatrix} \begin{pmatrix} \Delta u \\ \Delta\pi \\ \Delta u_\omega \\ \Delta\pi_\omega \\ \vdots \\ \Delta\gamma \end{pmatrix} = \begin{pmatrix} \widehat{b}_u \\ \widehat{b}_\pi \\ \widehat{b}_{u_\omega} \\ \widehat{b}_{\pi_\omega} \\ \vdots \\ \widehat{b}_\gamma \end{pmatrix}$$

What we have shown is the block bordered diagonal form that has the base case and one contingency block, but in general, under multiple contingencies, the above system will expand along the diagonal and border. The above system is still unacceptably large due to the significant number of control variables u and u_ω. As we stressed before, the number of active constraints is relatively small. Hence it would be computationally cheaper to eliminate the control variables from the above system. Once the control variables are eliminated, the final stage is a potentially small, bordered-block diagonal system of the form

$$\begin{pmatrix} C_0 & & & & & V_0^T \\ & C_1 & & & & V_1^T \\ & & C_2 & & & V_2^T \\ & & & \ddots & & \vdots \\ & & & & C_k & V_k^T \\ V_0 & V_1 & V_2 & \cdots & V_k & M \end{pmatrix} \begin{pmatrix} \Delta\pi_1 \\ \Delta\pi_2 \\ \Delta\pi_3 \\ \vdots \\ \Delta\pi_k \\ \Delta\gamma \end{pmatrix} = \begin{pmatrix} r_0 \\ r_1 \\ r_2 \\ \vdots \\ r_k \\ r_\gamma \end{pmatrix} \tag{4.3}$$

to be solved in the inner loop. Here the block matrix C_0 corresponds to the base case, the blocks C_ω, $\omega = 1, \ldots K$, correspond to the each of the contingency cases, and the bordering blocks V_k arise from the generator ramping constraints that couple the sub-problems.

Potentially, each diagonal block in (4.3) is as large as the number of all the line flow and control variable constraints in a single case; (4.3) could be enormous! But constraint relaxation limits the entries in each $\Delta\pi_\omega$ to just the constraints active for contingency ω, a number which is typically quite small compared with the size of the base case problem. A method for solving the above bordered block diagonal (also called multistage) system is suggested in [47], with the caution that

in this formulation, the block matrices C_k usually differ in size. The first $k+1$ equations have the form:

$$C_\omega \Delta\pi_\omega + V_\omega^T \Delta\gamma = r_\omega$$

from which $\Delta\pi_\omega$ can be expressed

$$\Delta\pi_\omega = C_\omega^{-1}\left(r_\omega - V_\omega^T \Delta\gamma\right) \tag{4.4}$$

The last equation in the matrix equation (4.3) is:

$$\sum_{\omega=0}^{k} V_\omega \Delta\pi_\omega + M\Delta\gamma = r_\gamma$$

After substituting $\Delta\pi_\omega$, last equation becomes:

$$\left(M - \sum_{\omega=0}^{k} V_\omega C_\omega^{-1} V_\omega^T\right)\Delta\gamma = r_\gamma - \sum_{\omega=0}^{k} V_\omega C_\omega^{-1} r_\omega \tag{4.5}$$

In order to solve (4.5) for $\Delta\gamma$, we have to factor each diagonal block C_ω as:

$$C_\omega = U_\omega^T D_\omega U_\omega$$

The computational steps in computing $\Delta\gamma$ are

$$V_\omega C_\omega^{-1} V_\omega^T = V_\omega U_\omega^{-1} D_\omega^{-1} U_\omega^{-T} V_\omega^T = K_\omega^T D_\omega^{-1} K_\omega$$

where $K_\omega = U_\omega^{-T} V_\omega^T$ is calculated by column fast-forward substitution. Also,

$$V_\omega C_\omega^{-1} r_\omega = V_\omega U_\omega^{-1} D_\omega^{-1} U_\omega^{-T} r_\omega = K_\omega^T D_\omega^{-1} \bar{r}_\omega$$

where the term $\bar{r}_\omega = U_\omega^{-T} r_\omega$ is calculated by forward substitution. Therefore, after this factorizations, (4.5) can be written as

$$\left(M - \sum_{\omega=0}^{k} K_\omega^T D_\omega^{-1} K_\omega\right)\Delta\gamma = r_\gamma - \sum_{\omega=0}^{k} K_\omega^T D_\omega^{-1} \bar{r}_\omega$$

$\Delta\gamma$ can be found from this equation. Now we can go back to (4.4) to calculate $\Delta\pi_\omega$ using the following procedure:

$$C_\omega \Delta\pi_\omega = r_\omega - V_\omega^T \Delta\gamma = \widetilde{r}_\omega$$

Since we already factored C_ω, we have

$$U_\omega^T D_\omega U_\omega \Delta\pi_\omega = \widetilde{r}_\omega$$

If we define

$$z = D_\omega U_\omega \Delta\pi_\omega$$

then z can be found from

$$U_\omega^T z = \widetilde{r}_\omega$$

by forward substitution and finally $\Delta\pi_\omega$ is calculated from

$$U_\omega \Delta\pi_\omega = D_\omega^{-1} z$$

by backward substitution. Then the algorithm calculates the rest of the unknowns iteratively.

These are the major steps in the nonlinear SQP CCOPF algorithm. The computation of the cost of multiple contingencies even with this compact formulation can be prohibitively expensive. For a 1,000-line network, the number of all possible double-line contingency scenarios is close to 5 million. The core question is how to choose the multiple contingencies to consider in order to obtain an accurate cost approximation. Stochastic modeling based on Monte Carlo methods is an attractive approach to a practical answer for such high dimension problems.

4.1.2 Importance Sampling

Importance Sampling- Basic Idea

To illustrate the basic idea of importance sampling we will discuss it first in its basic form, using it to approximately calculate the value of an integral. A more detailed introduction to general importance sampling can be found in references on Monte Carlo methods, [82] and [37]. that we used.

Let us consider a function $f(x)$ defined over the interval D and let us compute approximately an integral

$$I = \int_D f(x)dx$$

An underlying assumption is that in the above case the integrand is beyond our power of either theoretical integration or quadrature formulas, which is the case with a multidimensional integrand, where the variable of interest $x \in \mathbb{R}^k$. The idea is to calculate the above integral approximately as an expectation of a continuous random variable.

A naive Monte Carlo method would estimate I based on the independent identically distributed random samples $x^{(1)}, \ldots, x^{(N)}$ drawn uniformly from D. An approximation to I in that case can be obtained as:

$$I \approx \widehat{I}_N = \frac{1}{N} \sum_{j=1}^{N} f(x^{(j)})$$

The method called *importance sampling* proposed by Marshall in [55] provides a much better estimate. Suppose we could generate random samples $x^{(1)}, \ldots, x^{(N)}$ from a nonuniform distribution that puts more probability mass in the "important" parts of the sample space D. Let us explain the basic idea. Without loss of generality, let us assume the simple case where $D = [0, 1]$. In order to perform importance sampling, we first select a function $g(x)$ defined over the same interval as the integral that we want to calculate that satisfies two probability density function conditions:

1. The function $g(x)$ is positive inside $[0, 1]$
2. The integral of $g(x)$ over the whole interval $[0, 1]$ is equal to 1

$$\int_0^1 g(x)dx = 1$$

Then $g(x)$ is a density function for $0 \leq x \leq 1$, and we can calculate I as:

$$I = \int_0^1 f(x)dx = \int_0^1 \frac{f(x)}{g(x)} g(x)dx$$

If ξ is a random number sampled from the distribution $g(x)$ then we define the random variable

$$\eta = \frac{f(\xi)}{g(\xi)}$$

whose expectation is

$$E\{\eta\} = E\left\{\frac{f(\xi)}{g(\xi)}\right\} = \int_0^1 \frac{f(x)}{g(x)} g(x)dx = I$$

Now let us consider N independent, identically distributed random variables ξ_1, ξ_2, ..., ξ_N. According to the central limit theorem, for sufficiently large N, one can estimate the integral I by means of the unbiased estimator

$$I \approx \frac{1}{N} \sum_{j=1}^{N} \frac{f(\xi_j)}{g(\xi_j)}$$

which has a variance

$$\sigma^2_{f/g} = \int_0^1 \left(\frac{f(x)}{g(x)} - I\right)^2 g(x)dx$$

By proper choice of $g(x)$, one can theoretically reduce the variance substantially, well beyond that obtained using independent samples. In practice, successful importance sampling depends on the efficient choice of the importance sampling density $g(x)$. Theoretically, a candidate that produces zero variance is $I \cdot f(x)$, but its practical value is very low, since in order to select it we have to know I, the value that we want to estimate. Realistically, one may hope to find a good "candidate" $g(x)$ that follows the shape of $f(x)$ as much as possible or which will sample more in the regions where the value of $f(x)$ is high. However, generating random numbers from such a distribution can be a real challenge.

Importance Sampling in Stochastic OPF

The approach presented in this section follows the derivation in [46]. We present a general method applicable to multiple contingencies of any type. We just showed that importance sampling is a variance reduction technique which usually performs well with reasonable sample sizes. The objective in importance sampling is to concentrate the distribution of the sample points in the parts of the state space that are of most "importance" instead of spreading them out evenly.

A multiple-contingency state is modeled by a random vector v,

$$v = \begin{pmatrix} v_1 & v_2 & \dots & v_n \end{pmatrix}^T$$

where n is the number of independent random variables which could be line status, generator availability uncertainties, etc. If line contingencies are studied, n is the number of lines, and each of the entries v_i denotes line status. From the perspective of the sample space, line uncertainties are simpler to study than other kinds of uncertainties, since the state has only two possible realizations, in service or out of service. Therefore, v can have realization v^ω with corresponding probability $p(v^\omega)$, $\omega \in \Omega$, where Ω is the set of all possible contingency realizations. The number of all possible scenarios even for a modest order of multiple contingencies is not practically solvable. The operating cost function $c(x, u, v^\omega)$ depends on the state vector x, the control vector u, and the random vector v^ω, which represents the uncertainties. For simplicity of notation, we will denote the cost function by $c(v^\omega)$ to emphasize its stochastic character, which is crucial in this section.

Consider a random line outage scenario with random vector v^ω, $\omega \in \Omega$ and $N = |\Omega|$. The cost of the random line outage scenario $c(v^\omega)$ is an independent random variable with expected value $\overline{C}$, which in our discrete case is

$$\overline{C} = \sum_{\omega \in \Omega} c(v^\omega) p(v^\omega)$$

As we have shown in the integral example, by applying naive Monte Carlo, an unbiased estimator of the mean $\overline{C}$ is:

$$\overline{C} \approx \overline{z} = \frac{1}{N} \sum_{\omega=1}^{N} c(v^{\omega})$$

whose variance depends on the sample size as $O(\sqrt{N})$ regardless of the dimensionality of v. The expected value will be the same if we calculate it as

$$\overline{C} = \sum_{\omega \in \Omega} \frac{c(v^{\omega}) p(v^{\omega})}{q^{\omega}} q^{\omega}$$

by introducing a new sampling probability density function q^{ω}.

Successful importance sampling, as discussed, requires selecting an importance sampling density q^{ω} so that the variance in the estimate is reduced. For these reasons, we want q^{ω} to be proportional to $c(v^{\omega})p(v^{\omega})$ and at the same time computationally inexpensive to find. A Monte Carlo importance sampling estimator of $\overline{C}$ can be then defined as

$$\overline{z} = \frac{1}{N} \sum_{\omega=1}^{N} r^{\omega}$$

where the new random variable is

$$r^{\omega} = \frac{c(v^{\omega}) p(v^{\omega})}{q^{\omega}}$$

Now we will show how a potential candidate for a successful sampling function q^{ω} can be found. Let us introduce the notion of the "incremental cost" of a single line contingency. A single-line contingency state can be defined as the vector

$$\begin{pmatrix} \tau_1 & \dots & \tau_{i-1} & v_i & \tau_{i+1} & \dots & \tau_n \end{pmatrix}$$

with a single random variable corresponding to the base case

$$\begin{pmatrix} \tau_1 & \dots & \tau_n \end{pmatrix}$$

The incremental cost is defined as the difference between the cost of the contingency case arising from v_i and the base case

$$M_i(v_i) = c(\tau_1 \quad \dots \quad \tau_{i-1} \quad v_i \quad \tau_{i+1} \quad \dots \quad \tau_n) - c(\tau) \tag{4.6}$$

with corresponding expected value

$$\overline{M} = E\left\{M_i(v_i^{\omega})\right\} = \sum_{i=1}^{n} \left[c(\tau_1 \quad \dots \quad \tau_{i-1} \quad v_i \quad \tau_{i+1} \quad \dots \quad \tau_n) - c(\tau)\right] p_i^{\omega}$$

When line outages are studied, the expectation $\overline{M}$ simplifies to $\overline{M} = M_i$, since v_i can have only one different outcome than assumed. Since the incremental cost is proportional to the respective contingency cost (i.e., $M_i(v_i) \sim c(v^\omega)$), the expected value of the random outage scenario can be written as

$$\begin{aligned} \overline{C} &= \sum_{\omega\in\Omega} c(v^\omega)\frac{\overline{M}}{M(v^\omega)}\frac{M(v^\omega)}{\overline{M}}p(v^\omega) \\ &= \sum_{\omega\in\Omega} \overline{M}\,\underbrace{\frac{c(v^\omega)}{M(v^\omega)}}_{\text{new r.v.}}\underbrace{\frac{M(v^\omega)}{\overline{M}}p(v^\omega)}_{\text{new distribution}} \end{aligned}$$

or as the expectation

$$\overline{C} = \overline{M}E\Big\{\frac{c(v^\omega)}{M(v^\omega)}\Big\}$$

therefore the new random variable

$$F(v^\omega) = \overline{M}\frac{c(v^\omega)}{M(v^\omega)}$$

is distributed according to probability density function

$$q^\omega = \frac{M(v^\omega)}{\overline{M}}p(v^\omega)$$

For a particular network structure, we can calculate the base and all single-contingency OPF solutions in order to form an additive approximation of the cost function under multiple contingencies:

$$c(v) \approx c(\tau) + \sum_{i=1}^{n} M_i(v_i^\omega) \tag{4.7}$$

where M_i is the incremental cost of the single-line contingency, v_i represents a line outage scenario with probability p_i^ω, and $c(\tau)$ is the cost of the base case. The incremental cost is not too expensive to compute since we have to find one base case OPF solution and the n solutions of the single-line outage scenarios. The CCOPF formulation (4.3) shows that each single-line contingency will contribute to that set of equations one bordered diagonal block (one additional dimension beside the base case). In other words, we have to solve n one-dimensional CCOPF scenarios instead of one n-dimensional case.

The expected value of the cost (4.7) for the multiple contingency cases can be expressed in the following form:

$$E\Big\{c(v^\omega)\Big\} = c(\tau) + \sum_{i=1}^{n}\overline{M}\sum_{\omega\in\Omega}F(v^\omega)q_i^\omega\prod_{\substack{j=1\\ j\neq i}}^{n}p_j(v^\omega) \tag{4.8}$$

where

$$F(v^{\omega}) = \frac{c(v^{\omega}) - c(\tau)}{\sum_{i=1}^{n} M_i(v_i^{\omega})}$$

$$q^{\omega} = \frac{p_i(v^{\omega}) M_i(v_i^{\omega})}{\overline{M}}$$

Equation (4.8) can be interpreted as the sum of a constant term and n expectations. To describe the sampling scheme, partition the sample space Ω into n subspaces Ω_i,

$$\bigcup_{i=1}^{n} \Omega_i = \Omega$$

each of size n_i, corresponding to each line; assign each multi-line contingency to only one partition. Therefore each line i will be represented in double-line contingencies with weight n_i according to its incremental "importance"

$$n_i = \frac{M_i}{\overline{M}} N$$

The second component of the double-line outage in the subset Ω_i will be sampled according to the prescribed density function. In our case, since we do not have any *a priori* knowledge, it will be uniformly distributed among all other lines $j = 1, \ldots, n, \quad j \neq i$. Therefore, for each n_i, the i^{th} sum in (4.8) can be estimated by

$$\mu_i = \frac{1}{n_i} \sum_{j=1}^{n_j} F(v^j)$$

Finally, the estimated expected value of the double (in the general case, multiple) contingency can be written as:

$$\overline{z} = c(\tau) + \sum_{i=1}^{n} M_i \mu_i \tag{4.9}$$

4.1.3 Numerical example

The importance sampling technique coupled with the CCOPF formulation was tested on the IEEE 14-bus network. The ramp-rate constraints coefficient Δ was modeled as 10% of the generating capacity of each generator. This example tested $n - 2$ contingency cases. Since the IEEE 14-bus network has 20 lines, the number of all possible combinations for $n - 2$ contingency cases is 190. Since this is still a manageable number for our formulation, we found the exact cost of hedging

against all 190 cases and compared it with the estimated cost (4.9) obtained using importance sampling with three different sample sizes ($N = |\Omega|$). In the table, both the cost of the universe of all contingencies (*i.e.* all 190 second contingencies) and the estimated cost are normalized against the base case cost.

The total cost of hedging all 190 second contingency cases is $C_{n-2} = 1.315$ p.u. The fifth column of Table 4.1 shows the estimation error as a percent of C_{n-2}. This test case indicates that, as concluded in [46], importance sampling shows promise in the stochastic evaluation of multiple-contingency cases.

The implementation for large networks considering multiple contingencies will be the subject of future research. Our hope is that, as in other importance sampling applications described in [46], the method will be even more useful for investigating multiple contingencies on large networks. Future research will also incorporate load shedding into the formulation.

Table 4.1: Results for the IEEE 14-bus network test case

IEEE 14-bus network				
Case	**Sample size N**	**Sample size %**	**Estimated Normalized Cost function**	**Estimation Error %**
1	15	7.9	1.236	6.01
2	20	10.5	1.264	3.88
3	30	15.8	1.270	3.42

4.1.4 Conclusion

Evaluation of multiple contingencies is a challenging problem. The ultimate goal for any practical stochastic algorithm is to employ a sufficiently detailed model and to construct samples that emphasize the "important" part of the state space. In the formulation presented, a detailed model is obtained using nonlinear contingency-constrained OPF and a manageable sample size is achieved through importance sampling.

We have developed a mathematical formulation and tested it on the IEEE-14 bus network case. Results of the numerical example show that the expected costs obtained using importance sampling are close to the actual operating cost of accommodating the full universe of contingencies.

It is hoped that importance sampling-based methods will complement simulation methods in planning studies by filtering out from the large number of cases being studied those which require detailed scrutiny.

Chapter 5

A Formulation of the DC Contingency Constrained OPF for LMP Calculations

5.1 Introduction

Restructuring of the electric utility industry started with the unbundling of traditionally vertically integrated utility companies that provided generation, transmission, and distribution into independent, competitive commercial entities. Generating companies today sell electrical energy on the open market to which transmission companies have to provide open access. In the restructured industry, transmission companies are still treated as a monopoly, subject to regulation of the transmission tariffs they can charge for network access. The role of independent distribution companies is to provide low-voltage power to individual industrial, commercial and residential customers [43].

To ensure reliable, secure, and efficient operation of the power system, the Independent System Operator (ISO) entity has been established. The role of the ISO is

1. to be independent from market participants (i.e., electric utilities, generator owners, retailers);
2. to coordinate the use of the transmission system;
3. to operate the electric energy market.

With the restructuring of the electric utility industry, operation of the market has moved from being cost-based to bid-based. Under the Standard Market Design (SMD) issued by the Federal

Energy Commission (FERC) in 2002, the ISO as the central authority accepts supply and demand bids submitted by market participants (i.e., sellers and buyers). Once bids are submitted, the ISO performs a bid-based OPF to determine dispatch of the generation, calculate Locational Marginal Prices (LMP), and at the same time ensure secure and reliable operation of the power network.

Just as in the regulated industry, computer methods continue to play a major role in implementing the electricity market objective while ensuring secure system operations. A chart showing the inter-dependence between typical computer applications essential for successful energy market is depicted in Fig. 5.1.

Real-time snapshots of the system state are of paramount importance for market applications. In the electricity market environment a state estimator continues to serve the monitoring role essential for secure system operation. Its prominent role is to ensure that market modules are based on accurate on-line data and correct topology. The state estimation function utilized in the energy market is shown in Fig. 5.1. Only a robust and reliable state estimator can fulfill that need at all times. That segment of the problem is stressed in Chapter 2, where development of a robust estimator is discussed in detail.

The process of computing LMPs depicted in Fig. 5.1 is based on some form of contingency constrained OPF (CCOPF) and is decomposed into two stages. The information about the system status and selection of the bids subject to system constraints is performed by the LMP Preprocessor. The LMP Contingency Processor in Fig. 5.1 represent the contingency screening function. Its function is to identify efficiently active power flow binding inequalities. In this chapter we will present a novel algorithm in which this function can be efficiently performed through reduction of the underlying CCOPF problem. Ultimately, the LMP block in Fig. 5.1 computes the prices.

A Locational Marginal Price (LMP) at a particular node in the network is "the price of supplying an additional MW of load" at that bus. Or in other words, LMPs can be seen as the least expensive way of delivering one additional MW of electricity to a node in the network while respecting all system constrains.

The theory of LMPs, also called spot prices, was developed by Schweppe *et al.* in a few classical papers that preceded [75], where a comprehensive treatment of the subject can be found. The work by Hogan on contract networks in [39] is an important extension of Schweppe's idea.

The LMPs are obtained from the underlying OPF-based optimization problem. From a mathematical point of view, LMPs are derived from Lagrange multipliers or as a solution of the dual optimization problem. The traditional cost-based OPF, translates in the new market environment

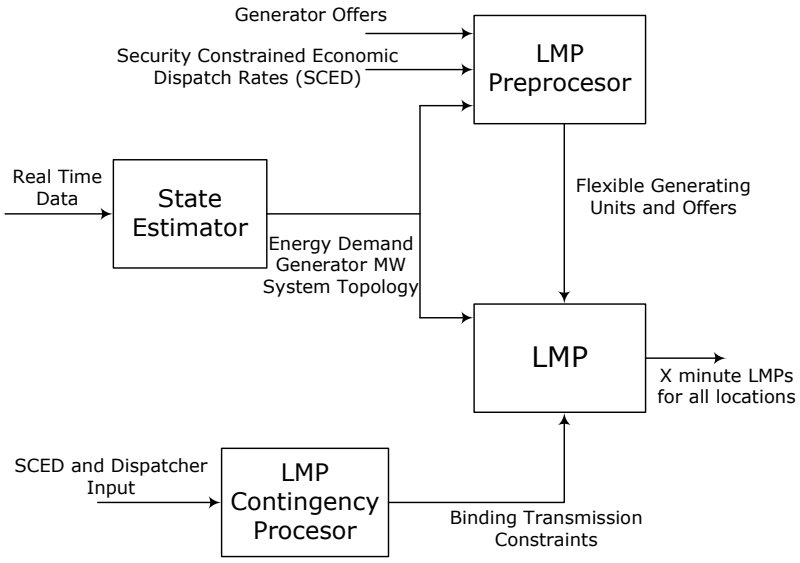

Figure 5.1: Typical Components of LMP Based Energy Market

into a bid-based OPF. Therefore, the problem objective is to find control settings that minimize the bid-based objective function constrained by meeting load demand while respecting all other constraints imposed on the problem. The resulting dispatch yields a set of market-clearing prices for energy market transactions and for transmission congestion charges. In a linear programming framework, bids are discrete bids, although in general other formulations of bid functions are possible [13].

The major factors affecting the LMP values are generator bid prices, the losses throughout the system, and transmission lines prone to congestion. Thus, each LMP has three components [4]

$$LMP = LMP^E + LMP^L + LMP^C$$

where:

LMP^E is the component due to the energy;

LMP^L is the component due to losses

LMP^C is the component due to congestion.

The energy component is the same throughout the system. In optimization language, the energy component is the Lagrange multiplier of the power balance equations at the reference bus (what we will define as α). The loss component varies and is usually small. If a lossless network model is used, as in our case, the loss component is neglected.

Transmission constraints are the cause of congestion. If line flow limits are binding, their effect

on the LMPs can be significant. Due to them the operator has to dispatch out-of-merit generation in order to meet the demand. Mathematically speaking, the congestion component is Lagrange multiplier of the binding line flow constraints; it will be defined soon as π_b in our algorithm. Therefore, transmission constraints contribute to the fluctuation of LMPs. The congestion component adds or subtracts from the LMP depending on whether power injection at the bus contributes to or alleviates congestion. These components will be much clearer once we derive the KKT conditions for the underlying optimization problem. We defer further discussion until then.

Under locational pricing, the cost of transmission congestion emerges as differences in energy prices between locations connected by a line whose flow hits its limit. The process that is currently in use by most ISO's for pricing congestion is based on the LMP-congestion component. Energy markets that adopted LMP-based congestion management agree that so far experience has been fairly successful. On a longer horizon, LMPs provide effective financial signals and incentives for locating new generation and transmission facilities which could provide further cost savings to energy consumers.

Although less accurate than full nonlinear OPF, a linear programming OPF formulation has been used almost exclusively in the LMP-based applications. Studies that examined the tradeoff between a full nonlinear OPF based on AC power flow against a linear OPF based on DC power flow have shown that results match fairly closely [66].

Linear programming OPF uses the DC power flow model. A favorable feature of the LP-based OPF is that it can handle many different contingencies in an efficient and computationally acceptable way. The cost of the computation in a linear OPF is substantially smaller than in a nonlinear OPF. A drawback of the DC power flow model is that it does not model power losses and is less accurate.

The development of a novel contingency constrained OPF algorithm suitable for market applications is the subject of the current chapter. We already discussed a closely related topic in Chapter 3, where the nonlinear CCOPF was used to estimate the cost of multiple contingencies. Since this chapter deals with energy market applications that have been governed by almost exclusively linear power flow models, our idea is to develop the CCOPF algorithm in the linear framework. The algorithm that we present efficiently calculates the dispatch, state and LMPs of the system under multiple contingencies.

The idea for problem decomposition that is used to develop the algorithm is based on work of Stott and Hobson in [86]. Once the KKT conditions for the original CCOPF problem have

been stated, the problem is decomposed into two stages. The first stage is a modified economic dispatch subproblem, whose solution allows efficient calculation of the system state and the LMP-congestion prices at the second stage. An interior point method is applied to the problem resulting in a bordered-block diagonal system for which an efficient solution exists. This formulation provides a framework to apply the importance sampling in order to obtain estimates of congestion charges under multiple contingencies.

5.2 Initial problem formulation

The objective in bid-based contingency constrained OPF is to find control settings that minimize the linear bid-based objective function

$$J = b^T u_0$$

Subject to the base case (pre-contingency) equality and inequality constraints

$$\begin{aligned} B_0\theta_0 + C_0 u_0 &= -p_L \\ F_0\theta_0 + G_0 u_0 &\leq f_0 \end{aligned}$$

as well as contingency constraints of the form

$$\begin{aligned} B_\omega\theta_\omega + C_\omega u_\omega &= -p_L \\ F_\omega\theta_\omega + G_\omega u_0 + H_\omega u_\omega &\leq f_\omega \\ \omega &= 1, \ldots, K \end{aligned}$$

The equality constraints are power balance equations at each bus in the network. The inequality constraints are limits imposed on the system components. Contingency constraints are incorporated either for corrective or preventive scheduling. In our case the corrective approach will be considered. Corrective control actions are modeled through ramp-rate constraints.

The general problem formulation is:

$$\begin{aligned} \text{Minimize} \quad & b^T u_0 \\ \text{Subject to:} \quad & B_0\theta_0 + C_0 u_0 = -p_L \\ & F_0\theta_0 + G_0 u_0 \leq f_0 \\ & B_\omega\theta_\omega + C_\omega u_\omega = -p_L \\ & F_\omega\theta_\omega + G_\omega u_0 + H_\omega u_\omega \leq f_\omega \\ & \omega = 1, \ldots, K \end{aligned} \tag{5.1}$$

The details of the formulation will be presented once the constraints used in the problem formulation are defined.

Nomenclature

$b \in \mathbb{R}^{n_g}$	is the bid vector;
$B \in \mathbb{R}^{n \times n}$	is the negative susceptance network matrix;
$\theta \in \mathbb{R}^{n}$	is the vector of bus angles (state variables);
$u \in \mathbb{R}^{n_u}$	is the vector of control variables;
$p_g \in \mathbb{R}^{n_g}$	is the vector of generator powers;
$p_l \in \mathbb{R}^{n_l}$	is the vector of nodal loads;
0	subscript that denotes variables or constraints associated with the base case;
ω	subscript that denotes variables or constraints associated with that contingency case;
n	number of network buses;
n_g	number of generators;
n_l	number of loads;
n_b	number of network branches, but in implementation the number of active line-flow constraints.

5.3 Modeling of Inequality Constraints

In our problem formulation we will have four types of inequality constraints. They are classified as follows:

- Transmission line flow limits (active power flow limits)
- Generator limits (lower and upper limits on real generation)
- Load-shedding limits
- Ramp-rate constraints

5.3.1 Transmission line flow limits using distribution factors

In DC power flow, active power line flow between nodes i and j is defined as

$$p_{ij} = \frac{1}{x_{ij}}(\theta_i - \theta_j)$$

where x_{ij} is the reactance of the line. We will define a vector p_{line} of all active power line flows, and a matrix $E \in \mathbb{R}^{n_b \times n}$ whose rows correspond to line flows and whose ij element has the form

$$E_{ij} = \frac{1}{x_{ij}}(e_i - e_j)^T$$

where e_i is the vector with all components equal to zero except for the i^{th} component, which is equal to 1. From the power balance equation,

$$B\theta = Kp_g - Mp_l$$

where:

$K \in \mathbb{R}^{n \times n_g}$	is the node-to-generator incidence matrix that has value 1 at position K_{ij} where i denotes a bus where generator j is connected;
$M \in \mathbb{R}^{n \times n_l}$	is the node-to-load incidence matrix that has value 1 at position M_{ij} where i denotes a bus where load j is connected

phase angles θ_i and θ_j can be obtained as

$$\begin{aligned} \theta_i &= e_i^T B^{-1}(Kp_g - Mp_l) \\ \theta_j &= e_j^T B^{-1}(Kp_g - Mp_l) \end{aligned}$$

Then, the line flow equation can be written as

$$p_{ij} = \frac{1}{x_{ij}}(e_i - e_j)^T B^{-1} K p_g - \frac{1}{x_{ij}}(e_i - e_j)^T B^{-1} M p_l$$

The vector p_{line} can be written as

$$p_{line} = EB^{-1}Kp_g - EB^{-1}Mp_l$$

where a matrix of so-called distribution factors can be defined as

$$F_b = EB^{-1}$$

Since the DC OPF problem requires LU factorization of B, distribution factors can be calculated at the cost of a two step forward/backward substitution. The first step is to find R^T by solving the equation

$$U^T R^T = E^T$$

via column-by-column forward substitution and the second is finding F_b^T from

$$L^T F_b^T = R^T$$

via column-by-column backward substitution. It is worthwhile to note that matrix F_b is non-sparse. Using distribution factors, line limit inequality constraints can be stated as

$$\overline{F}_b p_g - \widetilde{F}_b p_l \leq f_b$$

where

$$\begin{aligned} \overline{F}_b &= F_b K \\ \widetilde{F}_b &= F_b M \end{aligned}$$

Among the favorable properties of the DC OPF-based applications is one related to updating the system matrix B when the network is subject to contingencies. Since efficient contingency calculation is of particular interest in the development of the algorithm, we will show the computational steps for recomputing distribution factors of the network subject to line contingencies. When multiple (i.e., k-line) contingencies are considered, modifications to matrix B can be represented using U and V matrices in $\mathbb{R}^{n \times k}$. The general Sherman-Morrison-Woodbury formula [35] writes the inverse of $(B + UV^T)$ as

$$(B + UV^T)^{-1} = B^{-1} - B^{-1}U(I + U^T B^{-1} U)^{-1} V^T B^{-1}$$

which allows efficient recalculation of distribution factors.

When single contingencies are considered, the new B matrix, denoted as B_c, can be expressed as a rank-one modification:

$$B_c = B + uv^T$$

The updating procedure is a very important part of designing a computationally efficient algorithm. Using the rank-one Sherman-Morrison-Woodbury formula, B_c^{-1} can be written as

$$B_c^{-1} = (B + uv^T)^{-1} = B^{-1} - \frac{1}{1 + v^T B^{-1} u}(B^{-1}u)(v^T B^{-1})$$

Let us define

$$\gamma = \frac{1}{1 + v^T B^{-1} u}$$

For efficient solution, write

$$v^T B^{-1} u = v^T U^{-1} L^{-1} u = \bar{v}^T \bar{u}$$

where $\bar{v}$ is calculated from $U^T \bar{v} = v$ via fast-forward substitution, and $\bar{u}$ is calculated from $L\bar{u} = u$, also by fast-forward substitution. Thus,

$$B_c^{-1} = B^{-1} - \gamma \cdot \widetilde{u}\,\widetilde{v}^T \qquad \text{where} \qquad \gamma = \frac{1}{1 + \bar{v}^T \bar{u}}$$

and $\widetilde{v}$ is calculated from $U\widetilde{u} = \bar{u}$ via fast-backward substitution, and $\widetilde{v}$ is calculated from $L^T\widetilde{v} = \bar{v}$, also by fast-backward substitution. Therefore, the distribution factors for each contingency can be found by

$$F_b^c = F_b - \gamma \cdot E\widetilde{u}\,\widetilde{v}^T$$

5.3.2 Generator output limits

Generator output limits are constrained between

$$p_g^{min} \leq p_g \leq p_g^{max}$$

where

p_g^{max} is the maximum generation limit as determined by its rating;

p_g^{min} is the minimum generation limit, usually dependent on boiler stability and not necessarily zero.

For modeling purposes, we split each double-sided limit into two inequalities

$$\begin{aligned}
p_{g1} &\leq p_{g1}^{max} \\
p_{g2} &\leq p_{g2}^{max} \\
&\vdots \\
p_{gng} &\leq p_{gng}^{max} \\
-p_{g1} &\leq -p_{g1}^{min} \\
-p_{g2} &\leq -p_{g2}^{min} \\
&\vdots \\
-p_{gng} &\leq -p_{gng}^{min}
\end{aligned}$$

written in matrix form as

$$\begin{pmatrix} I_g \\ -I_g \end{pmatrix} p_g \leq \begin{pmatrix} p_g^{max} \\ -p_g^{min} \end{pmatrix}$$

$$F_g p_g \leq f_g$$

where I_g is the identity matrix of dimension $(n_g \times n_g)$.

5.3.3 Load shedding limits

Load shedding is included in both the objective function and in the constraint set. In the past, high cost has been assigned to the load shedding variables so that they are adjusted only as a last resort when no other solution can be achieved. Load shedding in today's market is tailored to customers' needs. By assigning proper weights we can model customers' participation in the market dispatch, especially if provided with forecasts of price information.

There are two alternatives for including load in the dispatch:

- voluntary - where customers agrees to adapt their demand to meet utility needs under uncertainty or during a period of high electricity price (congestion) or generation shortage;
- involuntary - by assigning very high weights and using load shedding.

Our formulation will allow this choice through the assignment of appropriate load weights c_i in the weight vector c. Load shedding limits represent the amount of load shed, generally bounded between 0 and the actual load $p_{l_i}^0$,

$$0 \leq p_{l_i} \leq p_{l_i}^0$$

which we write as

$$\begin{pmatrix} I_l \\ -I_l \end{pmatrix} p_l \leq \begin{pmatrix} p_l^0 \\ 0 \end{pmatrix}$$

$$F_l p_l \leq f_l$$

where I_l is the identity matrix of dimension $(n_l \times n_l)$

5.3.4 Ramp-rate constraints

Corrective control actions produce lower cost than preventive methods that are more conservative. In preventive methods contingency constraints are imposed in the base case and corrective actions are not allowed. One has to solve the base case such that a feasible operating state is achieved without considering the systems' corrective actions.

Corrective control actions involve changing the control variables of the system in response to a contingency occurrence within prespecified limits. This process is also known as post contingency corrective rescheduling. The underlying assumption is that rescheduling of the plant can be done within a maximum increment of Δ_i up or down.

General ramp-rate constraints are of the form

$$\underline{\Delta} \leq u - u^{\omega} \leq \overline{\Delta}$$

In our algorithm the control variables subject to ramp-rate constraints are active power generation.

$$\underline{\Delta} \leq p_g - p_{g\omega} \leq \overline{\Delta}$$

By replacing double-sided constraints with two sets of inequalities, as we have done before, one gets

$$H_0 p_g + H_\omega p_{g\omega} \leq \Delta$$

where

$$H_0 = \begin{pmatrix} I_g \\ -I_g \end{pmatrix}, \qquad H_\omega = \begin{pmatrix} -I_g \\ I_g \end{pmatrix} \qquad \text{and} \qquad \Delta = \begin{pmatrix} \overline{\Delta}_1 \\ \vdots \\ \overline{\Delta}_n \\ -\underline{\Delta}_1 \\ \vdots \\ -\underline{\Delta}_n \end{pmatrix}$$

where I_g is the identity matrix of dimension $(n_g \times n_g)$.

5.4 An Interior Point Solution Algorithm

The algorithm that we will derive in this section is based on an idea of Stott and Hobson in [86], which is that the linear programming formulation can be reduced to a smaller subproblem by

elimination of the phase angles and the Lagrange multipliers corresponding to the power balance equations.

In what follows, we adopt Stott's very elegant approach but solve the problem using an interior point method, prove some very interesting observation along the way, and extend the formulation to account for multiple contingencies.

The network power balance equation

$$B\theta = Kp_g - Mp_l$$

has to be decomposed due to the singularity of the network susceptance matrix B. To do this we impose the reference bus[3] equality constraint explicitly in the original set of power balance equations and treat separately its power balance equation. The corresponding modified power balance equation is

$$B'\theta = K'p_g - M'p_l$$

where B' is the modification of B in which its first row is replaced with vector e_1^T; the first rows of the incidence matrices K and M are zeroed out in order to obtain K' and M'. This modification reflects the constraint that the angle at the reference bus is equal to $0°$.

The power balance equation for the reference bus is treated separately and can be extracted from the initial set of power balance equations by premultiplying by e_1^T:

$$e_1^T B\theta = e_1^T Kp_g - e_1^T Mp_l$$

Therefore, the problem formulation is

$$\begin{aligned}
\text{Minimize} \quad & b^T p_g + c^T p_l \\
\text{Subject to} \quad & B'\theta - K'p_g + M'p_l = 0 \\
& e_1^T B\theta - e_1^T Kp_g + e_1^T Mp_l = 0 \\
& E\theta \leq f_b \\
& F_g p_g \leq f_g \\
& F_l p_l \leq f_l
\end{aligned}$$

The first step in the interior point method solution process is to convert inequality constraints to

[3] In this chapter the first bus in the network denotes the reference bus

equality constraints by introducing slack variables s_b, s_g and s_l:

$$\begin{aligned}
\text{Minimize} \quad & b^T p_g + c^T p_l \\
\text{Subject to} \quad & B'\theta - K' p_g + M' p_l = 0 \\
& e_1^T B\theta - e_1^T K p_g + e_1^T M p_l = 0 \\
& E\theta - f_b + s_b = 0 \\
& F_g p_g - f_g + s_g = 0 \\
& F_l p_l - f_l + s_l = 0
\end{aligned}$$

The nonnegativity of the slack variables is enforced by appending a logarithmic barrier function of the form

$$\mu\Big[\sum_{i=1}^{n_b} \ln s_b + \sum_{i=1}^{n_g} \ln s_g + \sum_{i=1}^{n_l} \ln s_l\Big]$$

The problem Lagrangian is given by

$$\begin{aligned}
\mathcal{L} = & \, b^T p_g + c^T p_l \\
& + \lambda^T \Big[B'\theta - K' p_g + M' p_l\Big] \\
& + \alpha\Big[e_1^T B\theta - e_1^T K p_g + e_1^T M p_l\Big] \\
& + \pi_b^T \Big[E\theta - f_b + s_b\Big] \\
& + \pi_g^T \Big[F_g p_g - f_g + s_g\Big] \\
& + \pi_l^T \Big[F_l p_l - f_l + s_l\Big] \\
& - \mu\Big[\sum_{i=1}^{n_b} \ln s_b + \sum_{i=1}^{n_g} \ln s_g + \sum_{i=1}^{n_l} \ln s_l\Big]
\end{aligned}$$

where the corresponding Lagrange multipliers in the LMP framework can be interpreted as:

α	is the energy component of the LMPs;
$\lambda(2:n)$	is the vector of LMPs;
π_b	is the vector of (shadow) *congestion price* for the line limit constraints

The KKT first-order necessary conditions

$$\frac{\partial \mathcal{L}}{\partial p_g} = b - K'^T\lambda - \alpha K^T e_1 + F_g^T \pi_g = 0 \tag{5.2}$$

$$\frac{\partial \mathcal{L}}{\partial p_l} = c + M'^T\lambda + \alpha M^T e_1 + F_l^T \pi_l = 0 \tag{5.3}$$

$$\frac{\partial \mathcal{L}}{\partial \theta} = B'^T\lambda + B^T e_1 \alpha + E^T \pi_b = 0 \tag{5.4}$$

$$\frac{\partial \mathcal{L}}{\partial \lambda} = B'\theta - K' p_g + M' p_l = 0 \tag{5.5}$$

$$\frac{\partial \mathcal{L}}{\partial \alpha} = e_1^T B\theta - e_1^T K p_g + e_1^T M p_l = 0 \tag{5.6}$$

$$\frac{\partial \mathcal{L}}{\partial \pi_b} = E\theta - f_b + s_b = 0 \tag{5.7}$$

$$\frac{\partial \mathcal{L}}{\partial \pi_g} = F_g p_g - f_g + s_g = 0 \tag{5.8}$$

$$\frac{\partial \mathcal{L}}{\partial \pi_l} = F_l p_l - f_l + s_l = 0 \tag{5.9}$$

$$\frac{\partial \mathcal{L}}{\partial s_b} = \Pi_b S_b - \mu e = 0 \tag{5.10}$$

$$\frac{\partial \mathcal{L}}{\partial s_g} = \Pi_g S_g - \mu e = 0 \tag{5.11}$$

$$\frac{\partial \mathcal{L}}{\partial s_l} = \Pi_l S_l - \mu e = 0 \tag{5.12}$$

The fundamental equation for understanding the idea behind LMP-based congestion prices is equation (5.4). The λ's are LMPs that, in the absence of congestion (no binding limits, i.e., $\pi_b = 0$), are equal to α, which is an energy price component or the price at the reference bus. Therefore, in the absence of congestion, prices are the same throughout the system. Once a line constraint becomes binding, its corresponding Lagrange multiplier becomes nonzero (i.e., $\pi_b \neq 0$), and the LMPs undergo changes. A very interesting discussion of equation (5.4) can be found in Wu *et al.* [99].

Reduction of the above system will be accomplished through elimination of λ and θ from the set of KKT conditions. Vectors λ and θ can be expressed from equations (5.4) and (5.5), respectively, as

$$\theta = B'^{-1} K' p_g - B'^{-1} M' p_l$$

$$\lambda = -B'^{-T} E^T \pi_b - B'^{-T} B^T e_1 \alpha$$

Substituting these expressions in the rest of the system results in

$$b + K'^T B'^{-T} E^T \pi_b + \alpha \Big[K'^T B'^{-T} B^T - K^T \Big] e_1 + F_g^T \pi_g = 0 \tag{5.13}$$

$$c - M'^T B'^{-T} E^T \pi_b - \alpha \Big[M'^T B'^{-T} B^T - M^T \Big] e_1 + F_l^T \pi_l = 0 \tag{5.14}$$

$$e_1^T B \Big[B'^{-1} K' p_g - B'^{-1} M' p_l \Big] = e_1^T K p_g - e_1^T M p_l \tag{5.15}$$

$$E B'^{-1} K' p_g - E B'^{-1} M' p_l - f_b + s_b = 0 \tag{5.16}$$

$$F_g p_g - f_g + s_g = 0 \tag{5.17}$$

$$F_l p_l - f_l + s_l = 0 \tag{5.18}$$

$$\Pi_g S_g - \mu e = 0 \tag{5.19}$$

$$\Pi_b S_b - \mu e = 0 \tag{5.20}$$

$$\Pi_l S_l - \mu e = 0 \tag{5.21}$$

In order to simplify further, we will show that the following two equations hold:

$$K'^T B'^{-T} B^T e_1 - K^T e_1 = \bar{e} \quad \text{where} \quad \bar{e} = (1 \dots 1)^T \in \mathbb{R}^{n_g}$$

$$M'^T B'^{-T} B^T e_1 - M^T e_1 = \tilde{e} \quad \text{where} \quad \tilde{e} = (1 \dots 1)^T \in \mathbb{R}^{n_l}$$

One may recall that K is the node-to-generator incidence matrix, each of whose columns has exactly one element equal to one and the rest of the elements are zero. K' is the matrix K modified in such a way that its first row is zeroed out. Accordingly, two cases are considered

1. There is no generator connected to the reference bus.

   In this case, each row of K'^T has exactly one element equal to 1 and the first column is the zero vector. Also the product $K'e_1$ is the zero vector

$$K'^T = \begin{pmatrix} 0 & \times & \times & \cdots & \times \\ 0 & \times & \times & \cdots & \times \\ 0 & \times & \times & \cdots & \times \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \times & \times & \cdots & \times \end{pmatrix} \qquad \text{and} \qquad K'e_1 = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

2. Generator j, $(j \neq 1)$, is connected to the reference bus.

   In this case the j^{th} row of K'^T is a zero vector, while the j^{th} element of vector $K'e_1$ will have

value 1 and zero everywhere else.

$$K'^T = \begin{pmatrix} 0 & \times & \times & \cdots & \times \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \vdots & 0 \\ 0 & \times & \times & \cdots & \times \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \times & \times & \cdots & \times \end{pmatrix} \qquad \text{and} \qquad K'e_1 = \begin{pmatrix} 0 \\ \vdots \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

According to Theorem B.4. on page 143 in the Appendix B, the product $B'^{-T}B^T$ is equal to

$$B'^{-T}B^T = \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 \\ -1 & 1 & 0 & \cdots & 0 \\ -1 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -1 & 0 & 0 & \cdots & 1 \end{pmatrix}$$

With this matrix structure, one can easily show that whether or not a generator is connected to the reference bus, one gets

$$K'^T B'^{-T} B^T e_1 - K^T e_1 = \bar{e} \quad \text{where} \quad \bar{e} \in \mathbb{R}^{n_g}$$

In a similar way it can be shown that

$$M'^T B'^{-T} B^T e_1 = -\tilde{e} \quad \text{where} \quad \tilde{e} \in \mathbb{R}^{n_l}$$

Equation (5.15) can be rewritten as

$$e_1^T \Big[BB'^{-1}K' - K \Big] p_g = e_1^T \Big[BB'^{-1}M' - M \Big] p_l$$

From the above discussion it is straightforward to show that

$$e_1^T \Big[BB'^{-1}K' - K \Big] = -\bar{e}^T \quad \text{where} \quad \bar{e} \in \mathbb{R}^{n_g}$$

and also

$$e_1^T \Big[BB'^{-1}M' - M \Big] = -\tilde{e}^T \quad \text{where} \quad \tilde{e} \in \mathbb{R}^{n_l}$$

Therefore, the power balance equation for the reference bus (5.15), after elimination of the vector θ, becomes the system power balance equation

$$\bar{e}^T p_g = \tilde{e}^T p_l$$

One may recall that we already encountered the terms

$$EB'^{-1}K' = \overline{F}_b$$
$$EB'^{-1}M' = \widetilde{F}_b$$

as the distribution factors discussed on page 110.

Thus, the KKT conditions can be written in more compact form as:

$$\begin{aligned}
&b + \overline{F}_b^T \pi_b + \bar{e}\alpha + F_g^T \pi_g = 0 \\
&c - \widetilde{F}_b^T \pi_b - \widetilde{e}\alpha + F_l^T \pi_l = 0 \\
&\bar{e}^T p_g - \widetilde{e}^T p_l = 0 \\
&\overline{F}_b p_g - \widetilde{F}_b p_l - f_b + s_b = 0 \\
&F_g p_g - f_g + s_g = 0 \\
&F_l p_l - f_l + s_l = 0 \\
&\Pi_b S_b - \mu e = 0 \\
&\Pi_g S_g - \mu e = 0 \\
&\Pi_l S_l - \mu e = 0
\end{aligned} \tag{5.22}$$

This reduced system of KKT conditions can be seen as the KKT conditions of the following Lagrangian:

$$\begin{aligned}
\mathcal{L} = {} & b^T p_g + c^T p_l \\
& + \alpha \Big[\bar{e}^T p_g - \tilde{e}^T p_l \Big] \\
& + \pi_b^T \Big[\overline{F}_b p_g - \widetilde{F}_b p_l - f_b + s_b \Big] \\
& + \pi_g^T \Big[F_g p_g - f_g + s_g \Big] \\
& + \pi_l^T \Big[F_l p_l - f_l + s_l \Big] \\
& - \mu \Big[\sum_{i=1}^{n_b} \ln s_b + \sum_{i=1}^{n_g} \ln s_g + \sum_{i=1}^{n_l} \ln s_l \Big]
\end{aligned}$$

The corresponding reduced problem is

$$\begin{aligned} \text{Minimize} \quad & b^T p_g + c^T p_l \\ \text{Subject to} \quad & \bar{e}^T p_g = \tilde{e}^T p_l \\ & \overline{F}_b p_g - \widetilde{F}_b p_l \leq f_b \\ & F_g p_g \leq f_g \\ & F_l p_l \leq f_l \end{aligned} \tag{5.23}$$

One can recognize this problem as an economic dispatch problem with line limits imposed via distribution factors.

5.4.1 Solution of the reduced system

In this section we will discus how the reduced order system can be solved using an interior point method, The reduced KKT conditions (5.22) are nonlinear due to the last three complimentary slackness conditions. They are linearized as follows:

$$\Pi_g \Delta s_g + S_g \Delta \pi_g = \mu e - \Pi_g S_g e$$
$$\Pi_b \Delta s_b + S_b \Delta \pi_b = \mu e - \Pi_b S_b e$$
$$\Pi_l \Delta s_l + S_l \Delta \pi_l = \mu e - \Pi_l S_l e$$

Now express Δs_g, Δs_b, Δs_l as

$$\Delta s_g = \mu \Pi_g^{-1} e - s_g - \Pi_g^{-1} S_g \Delta \pi_g \tag{5.24}$$
$$\Delta s_b = \mu \Pi_b^{-1} e - s_b - \Pi_b^{-1} S_b \Delta \pi_b \tag{5.25}$$
$$\Delta s_l = \mu \Pi_l^{-1} e - s_l - \Pi_l^{-1} S_l \Delta \pi_l \tag{5.26}$$

and substitute them in the rest of the linearized system, which becomes

$$\begin{aligned} & F_g^T \Delta \pi_g + \overline{F}_b^T \Delta \pi_b + \alpha \bar{e} = r_1 \\ & F_l^T \Delta \pi_l - \widetilde{F}_b^T \Delta \pi_b - \alpha \tilde{e} = r_2 \\ & \bar{e}^T p_g - \tilde{e}^T p_l = 0 \\ & F_g p_g - D_g \Delta \pi_g = r_3 \\ & \overline{F}_b p_g - \widetilde{F}_b p_l - D_b \Delta \pi_b = r_4 \\ & F_l p_l - D_l \Delta \pi_l = r_5 \end{aligned} \tag{5.27}$$

where

$$\begin{aligned} r_1 &= -b - F_g^T \pi_g - \overline{F}_b^T \pi_b \\ r_2 &= -c - F_l^T \pi_l - \widetilde{F}_b^T \pi_b \\ r_3 &= f_g - \mu \Pi_g^{-1} e \\ r_4 &= f_b - \mu \Pi_b^{-1} e \\ r_5 &= f_l - \mu \Pi_l^{-1} e \end{aligned}$$

and

$$\begin{aligned} D_g &= \Pi_g^{-1} S_g \\ D_b &= \Pi_b^{-1} S_b \\ D_l &= \Pi_l^{-1} S_l \end{aligned}$$

The next step is to express the vectors $\Delta\pi_g$, $\Delta\pi_b$ and $\Delta\pi_l$ from the system (5.27) as

$$\Delta\pi_g = D_g^{-1} F_g p_g - D_g^{-1} r_3 \tag{5.28}$$

$$\Delta\pi_b = D_b^{-1} \overline{F}_b p_g - D_b^{-1} \widetilde{F}_b p_l - D_l^{-1} r_4 \tag{5.29}$$

$$\Delta\pi_l = D_l^{-1} F_l p_l - D_l^{-1} r_5 \tag{5.30}$$

Eliminating (5.28), (5.29) and (5.30) results in the matrix form

$$\begin{pmatrix} F_g^T D_g^{-1} F_g + \overline{F}_b^T D_b^{-1} \overline{F}_b & -\overline{F}_b^T D_b^{-1} \widetilde{F}_b & \bar{e} \\ -\widetilde{F}_b^T D_b^{-1} \overline{F}_b & F_l^T D_l^{-1} F_l + \widetilde{F}_b^T D_b^{-1} \widetilde{F}_b & -\tilde{e} \\ \bar{e}^T & -\tilde{e}^T & 0 \end{pmatrix} \begin{pmatrix} p_g \\ p_l \\ \alpha \end{pmatrix} = \begin{pmatrix} r_6 \\ r_7 \\ 0 \end{pmatrix} \tag{5.31}$$

where the right hand side terms are

$$\begin{aligned} r_6 &= r_1 + F_g^T D_g^{-1} r_3 + \overline{F}_b^T D_b^{-1} r_4 \\ r_7 &= r_2 + F_l^T D_l^{-1} r_5 - \widetilde{F}_b^T D_l^{-1} r_4 \end{aligned}$$

The pseudocode for a DC OPF algorithm based on this form is outlined in Algorithm 9.

5.5 Formulation of the DC Contingency Constrained OPF

The DC contingency constrained OPF problem may be formulated as a single optimization problem which includes a base case and a set of contingency cases coupled with ramp-rate constraints.

Algorithm 9 DCOPF algorithm

given an initial dispatch p_g

build initial F_g and F_l

initialize μ

while $\mu \geq \epsilon$ **do**

calculate p_g, p_l

calculate $\Delta\pi_g$, $\Delta\pi_b$ and $\Delta\pi_l$

calculate Δs_g, Δs_b, Δs_l

calculate step size

update $\Delta\pi$ and Δs vectors

update μ

end while

check for new violations

while new violations $\neq 0$ **do**

build new $\overline{F}_b$ and $\widetilde{F}_b$

% resolve the problem

initialize μ

while $\mu \geq \epsilon$ **do**

calculate p_g, p_l

calculate $\Delta\pi_g$, $\Delta\pi_b$ and $\Delta\pi_l$

calculate Δs_g, Δs_b, Δs_l

calculate step size

update $\Delta\pi$ and Δs vectors

update μ

end while

end while

calculate θ and λ

The mathematical formulation is as follows

$$\begin{aligned}
\text{Minimize} \quad & b^T p_g + c^T p_l \\
\text{Subject to} \quad & \bar{e}^T p_g = \tilde{e}^T p_l \\
& \overline{F}_b p_g - \widetilde{F}_b p_l \leq f_b \\
& F_g p_g \leq f_g \\
& F_l p_l \leq f_l \\
& \bar{e}^T p_{g\omega} = \tilde{e}^T p_l \\
& \overline{F}_{b\omega} p_{g\omega} - \widetilde{F}_{b\omega} p_l \leq f_{b\omega} \\
& F_{g\omega} p_{g\omega} \leq f_{g\omega} \\
& F_{l\omega} p_l \leq f_{l\omega} \\
& H_0 p_g + H_\omega p_\omega \leq \Delta \\
& \omega = 1, \ldots, K
\end{aligned}$$

Instead of deriving the full algorithm, we will just look at terms that will be affected by extending the problem to include contingencies. We know from the nonlinear CCOPF covered in Chapter 4 that each contingency case introduces a problem as large as the base case and that the base case and contingency cases are coupled via the ramp-rate constraints. Addition of ramp-rate constraints will expand certain terms in the base case KKT conditions and add appropriate blocks for each contingency case considered. Once the impact of the ramp-rate constraints upon the base case problem structure is examined, the pattern of the full linear CCOPF will emerge.

Addition of the ramp-rate constraint

$$H_0 p_g + H_\omega p_\omega \leq \Delta$$

to the base case will add the following terms to the base case problem Lagrangian

$$\begin{aligned}
\mathcal{L} = \cdots + & \pi_{r\omega}^T \Big[H_0 p_g + H_\omega p_\omega - \Delta + s_{r\omega} \Big] \\
& - \mu \sum_{\omega=1}^{K} \sum_{i=1}^{n_g} \ln s_{r\omega}
\end{aligned}$$

Those new terms will modify the following KKT condition

$$\frac{\partial \mathcal{L}}{\partial p_g} = b + F_g^T \pi_g + \overline{F}_b^T \pi_b + \alpha \bar{e} + \sum_{\omega=1}^{K} H_0^T \pi_{r\omega} = 0$$

as well as add two new KKT conditions

$$\frac{\partial \mathcal{L}}{\partial \pi_{r\omega}} = H_0 p_g + H_\omega p_{g\omega} - \Delta + s_{r\omega} = 0$$
$$\frac{\partial \mathcal{L}}{\partial s_{r\omega}} = \Pi_{r\omega} S_{r\omega} - \mu e = 0$$

where $S_{r\omega} = \text{diag}(s_{r\omega})$, $\Pi_{r\omega} = \text{diag}(\pi_{r\omega})$. The KKT conditions linearized around $\pi_{r\omega}$ and $s_{r\omega}$ are

$$F_g^T \Delta\pi_g + \overline{F}_b^T \Delta\pi_b + \alpha\bar{e} + \sum_{\omega=1}^{K} H_0^T \Delta\pi_{r\omega} = r_1' \tag{5.32}$$

$$H_0 p_g + H_\omega p_{g\omega} - \Delta + s_{r\omega} + \Delta s_{r\omega} = 0 \tag{5.33}$$

$$\Pi_{r\omega}\Delta s_{r\omega} + S_{r\omega}\Delta\pi_{r\omega} = \mu e - \Pi_{r\omega} S_{r\omega} e \tag{5.34}$$

For convenience we will define

$$r_1' \quad = \quad r_1 - \sum_{\omega=1}^{K} H_0^T \pi_{r\omega}$$

By expressing the incremental slack variable $\Delta s_{r\omega}$ from the linearized complementary slackness equation as

$$\Delta s_{r\omega} = \mu \Pi_{r\omega}^{-1} e - s_{r\omega} - \Pi_{r\omega}^{-1} S_{r\omega} \Delta\pi_{r\omega}$$

and substituting in (5.33) one gets

$$H_0 p_g + H_\omega p_{g\omega} - D_{r\omega}\Delta\pi_{r\omega} = r_{10\omega}$$

where

$$D_{r\omega} \quad = \quad \Pi_{r\omega}^{-1} S_{r\omega}$$
$$r_{10\omega} \quad = \quad \Delta - \mu \Pi_{r\omega}^{-1} e$$

Now $\Delta\pi_{r\omega}$ can be eliminated from

$$\Delta\pi_{r\omega} = D_{r\omega}^{-1} H_0 p_g + D_{r\omega}^{-1} H_\omega p_{g\omega} - D_{r\omega}^{-1} r_{10\omega} \tag{5.35}$$

After substituting $\Delta\pi_{r\omega}$ into (5.34) and a bit of algebra, the equation has the form

$$\Big[F_g^T D_g^{-1} F_g + \overline{F}_b^T D_b^{-1} \overline{F}_b + \sum_{\omega=1}^{K} H_0^T D_{r\omega}^{-1} H_0\Big] p_g - \overline{F}_b^T D_b^{-1} \widetilde{F}_b p_l + \lambda\bar{e} + \sum_{\omega=1}^{K} H_0^T D_{r\omega}^{-1} H_\omega p_{g\omega} = r_1'' \tag{5.36}$$

where

$$r_1'' = r_1' + \sum_{\omega=1}^{K} H_0^T D_{r\omega}^{-1} r_{10\omega}$$

which closes consideration of the base case with ramp-rate constraint appended.

The next stage is to consider the general form of the contingency part. As stated before, the KKT conditions for the contingency part of the problem are very similar to the base case, and all of them can be obtained from the base case consideration by appending the subscript ω. Due to the coupling constraints, only the $\frac{\partial \mathcal{L}}{\partial p_{g\omega}}$ condition requires special consideration. Therefore, $\frac{\partial \mathcal{L}}{\partial p_{g\omega}}$ has the form

$$\frac{\partial \mathcal{L}}{\partial p_{g\omega}} = \bar{e}\alpha_\omega + F_{g\omega}^T \pi_{g\omega} + \overline{F}_{b\omega}^T \pi_{b\omega} + H_\omega^T \pi_{r\omega} = 0$$

Using the same linearization process as in the base case leads to the final form

$$H_\omega^T D_{r\omega}^{-1} H_0 p_g + \left[F_{g\omega}^T D_{g\omega}^{-1} F_{g\omega} + \overline{F}_{b\omega}^T D_{b\omega}^{-1} \overline{F}_{b\omega} + H_\omega^T D_{r\omega}^{-1} H_\omega\right] p_{g\omega} - \overline{F}_{b\omega}^T D_{b\omega}^{-1} \widetilde{F}_{b\omega} p_{l\omega} + \alpha_\omega \bar{e} = r_{1\omega}''$$

where

$$r_{1\omega}'' = r_{1\omega}' + H_\omega^T D_{r\omega}^{-1} r_{10\omega}$$

The coupling between the base and the contingency cases is best seen if we represent all equations in block matrix form. The following compact form produces the well-known upper bordered-diagonal system, similar to the lower bordered-diagonal system obtained for the nonlinear CCOPF.

$$\begin{pmatrix} C_0 & V_1 & V_2 & \cdots & V_k \\ V_1^T & C_1 & & & \\ V_2^T & & C_2 & & \\ \vdots & & & \ddots & \\ V_k^T & & & & C_k \end{pmatrix} \begin{pmatrix} p_0 \\ p_1 \\ p_2 \\ \vdots \\ p_k \end{pmatrix} = \begin{pmatrix} r_0 \\ r_1 \\ r_2 \\ \vdots \\ r_k \end{pmatrix} \tag{5.37}$$

where each block has the structure

$$C_0 = \begin{pmatrix} C_{11} & C_{12} & \bar{e} \\ C_{21} & C_{22} & -\tilde{e} \\ \bar{e}^T & -\tilde{e}^T & 0 \end{pmatrix} \quad , \quad V_1 = \begin{pmatrix} C_{14} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad \text{and} \quad p_0 = \begin{pmatrix} p_g \\ p_l \\ \alpha \end{pmatrix}$$

The base-case block matrices are defined as:

$$C_{11} = F_g^T D_g^{-1} F_g + \overline{F}_b^T D_b^{-1} \overline{F}_b + \sum_{\omega=1}^{K} H_0^T D_{r\omega}^{-1} H_0$$
$$C_{12} = -\overline{F}_b^T D_b^{-1} \widetilde{F}_b$$
$$C_{14} = H_0^T D_{r\omega}^{-1} H_\omega$$
$$C_{21} = C_{12}^T$$
$$C_{22} = F_l^T D_l^{-1} F_l + \widetilde{F}_b^T D_b^{-1} \widetilde{F}_b$$

The coupling block matrices are defined as:

$$C_{41} = C_{14}^T = H_\omega^T D_{r\omega}^{-1} H_0$$

and the contingency block matrices are defined as:

$$C_{11}^\omega = F_{g\omega}^T D_{g\omega}^{-1} F_{g\omega} + \overline{F}_{b\omega}^T D_{b\omega}^{-1} \overline{F}_{b\omega} + H_\omega^T D_{r\omega}^{-1} H_\omega$$
$$C_{12}^\omega = -\overline{F}_{b\omega}^T D_{b\omega}^{-1} \widetilde{F}_{b\omega}$$
$$C_{21}^\omega = C_{12}^{\omega T}$$
$$C_{22}^\omega = F_{l\omega}^T D_{l\omega}^{-1} F_{l\omega} + \widetilde{F}_{b\omega}^T D_{b\omega}^{-1} \widetilde{F}_{b\omega}$$

5.5.1 Solution of the upper Bordered-diagonal system

Next a procedure for solving the bordered-diagonal system (5.37) will be outlined. Equations 2 to k have the same form and can be written as

$$V_\omega^T p_0 + C_\omega p_\omega = r_\omega \qquad \omega = 1, \ldots, K$$

Express p_ω as

$$p_\omega = C_\omega^{-1}(r_\omega - V_\omega^T p_0) \tag{5.38}$$

The first equation from (5.37) is

$$\sum_{\omega=1}^{k} V_\omega p_\omega + C_0 p_0 = r_0$$

which after substituting p_ω from (5.38) becomes

$$\left(C_0 - \sum_{\omega=1}^{k} V_\omega C_\omega^{-1} V_\omega^T \right) p_0 = r_0 - \sum_{\omega=1}^{k} V_\omega C_\omega^{-1} r_\omega \tag{5.39}$$

The first step in solving this equation is to factor each symmetric block matrix C_ω as

$$C_\omega = U_\omega^T D_\omega U_\omega$$

Then calculating the terms in the sum on the left-hand side as

$$V_\omega C_\omega^{-1} V_\omega^T = V_\omega U_\omega^{-1} D_\omega^{-1} U_\omega^{-T} V_\omega^T = K_\omega^T D_\omega^{-1} K_\omega$$

with K_ω calculated column-by-column via fast-forward substitution from

$$U_\omega^T K_\omega = V_\omega^T$$

In a similar way the terms on the right-hand side of the summation are calculated as

$$V_\omega U_\omega^{-1} D_\omega^{-1} U_\omega^{-T} r_\omega = K_\omega^T D_\omega^{-1} \bar{r}_\omega$$

where

$$\bar{r}_\omega = U_\omega^{-T} r_\omega$$

is calculated by forward substitution. Thus, equation (5.38) has the form

$$\left(C_0 - \sum_{\omega=1}^{k} K_\omega^T D_\omega^{-1} K_\omega \right) p_0 = r_0 - \sum_{\omega=1}^{k} K_\omega D_\omega^{-1} \bar{r_\omega}$$

from which p_0 can be found by performing LU factorization of the matrix

$$C_0 - \sum_{\omega=1}^{k} K_\omega^T D_\omega^{-1} K_\omega$$

Once p_0 is found, the p_ω's are calculated from equations 1 to k of the system (5.37)

$$U_\omega^T D_\omega U_\omega p_\omega = r_\omega - V_\omega^T p_0$$

where p_ω can be found by forward/backward substitution by first finding z from

$$U_\omega^T \cdot z = r_\omega - V_\omega^T p_0$$

via forward substitution and then p_ω from

$$U_\omega p_\omega = D_\omega^{-1} \cdot z$$

by backward substitution.

5.6 Importance sampling for LMP-based congestion prices

In practice, LMPs that respect the standard $N-1$ reliability criteria are obtained in the following way: the system operator identifies the worst single contingency and performs CCOPF with that contingency to obtain LMPs that meet standard reliability criteria. Finding single worst contingency is still a manageable job even for a large system. If one is interested in going beyond standard reliability criteria, it is an open question as to what to do. As we explained earlier, if we go one step further, the number of $N-2$ cases could be prohibitively large.

The real challenge is how to define schemes for the evaluation of multiple contingencies without considering all of them and still obtain an acceptable estimate of the relevant variables. A method based on probability is required to gain more insight into the cost of congestion. What we suggest is to find a valid sample space, similar to the one presented in Chapter 4, and apply the importance sampling algorithm. Experience suggests that such an algorithm will give a good estimate of the congestion prices under multiple contingencies.

Let us reiterate the basic ideas of importance sampling algorithm described in Chapter 4. The algorithm first assesses all single contingencies and finds their incremental cost (M_i), which is the difference between bid value of each contingency case (J_ω) and the base case (J). Then one finds the expected value of the incremental cost $\overline{M}$ for all single contingencies. One has to choose the size N of the sample space Ω for the multiple contingencies to be considered. Partition the sample space Ω into n_b subspaces Ω_i where $\bigcup_{i=1}^{n_b} \Omega_i = \Omega$, each of size n_i, corresponding to each line; assign each multi-line contingency to only one partition. Therefore, each line i will be represented in a double-line contingency with weight n_i according to its marginal "importance"

$$n_i = \frac{M_i}{\overline{M}} N$$

The second component (the second line in the double-line contingency) will be sampled randomly. Finally, the congestion price at each node is calculated according to:

$$\overline{\lambda} = \frac{1}{N} \sum_{k=1}^{N} \lambda_{\omega k}$$

The cost of security under multiple contingencies is estimated as

$$\overline{\lambda} = \frac{1}{N} \sum_{k=1}^{N} \lambda_{\omega k} - \alpha$$

The importance sampling algorithm for LMP-based congestion and cost of security estimation, using contingency constrained DC OPF as developed in this chapter is proposed in Fig. 5.2.

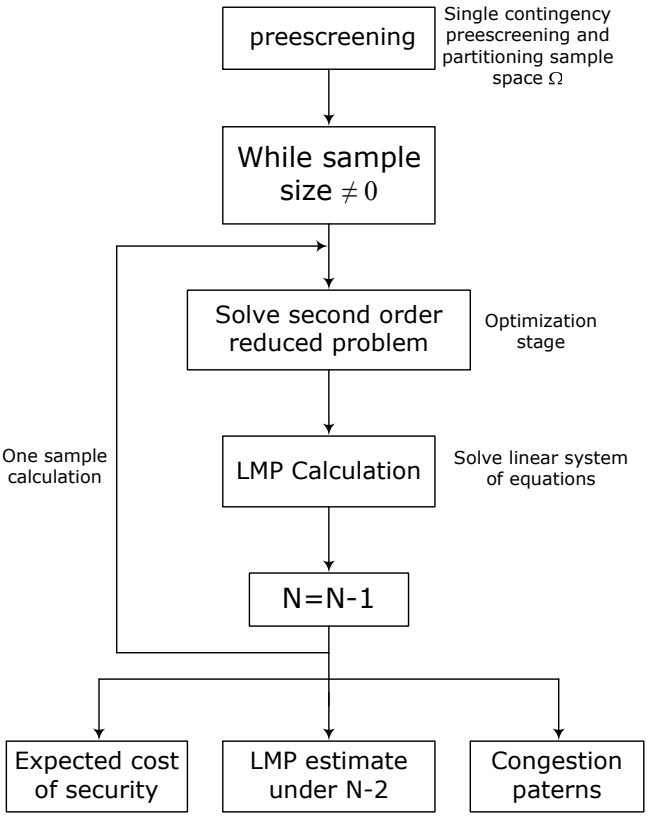

Figure 5.2: Importance sampling in contingency constrained DC OPF framework

Chapter 6

Conclusions and Future Work

6.1 Conclusions

The ability of the state estimator to achieve a high level of efficiency and numerical robustness is of paramount importance in today's eclectic utility industry. A robust algorithm must be *globally convergent* (convergent from any starting point), and able to solve in practice both well-conditioned and ill-conditioned problems.

This dissertation presents a new approach for solving power system state estimation based on a globally convergent modification of Newton's method using trust region methods (TRM). The performance of the TRM method was tested on the standard IEEE network cases and results are discussed thoroughly. A sound theoretical support as well as practical efficiency and robustness are the strong arguments supporting the trust region method to be applied in practical power system state estimators. The objective is to provide a more reliable and robust state estimator, which can successfully cope with all kinds of errors (bad data, topological, parameter) faced in power system models.

It is well known that Krylov subspace iterative methods are used to solve large sparse linear systems. Although it was not clear their potential on the power system state estimation problems. In presented research it has been found that LSQR method perform reliably when applied to solve PSSE. The LSQR method follows the same principle as CG, although it is much better suited for least-squares problems. The numerical simulations indicate that LSQR method is very competitive in robustness with classical QR factorization algorithm. Additional savings by reduction is number of floating point operations, no need for ordering, and ability to implement iterative methods using parallel computing, recommend Newton-LSQR method for practical implementations.

The dissertation presents SQP technique combined with the method of importance sampling in order to solve the stochastic OPF. The objective in importance sampling is to concentrate the random sample points in critical regions of the state space. In our case that means that single-line outages that cause the most "trouble" will be encountered more frequently in multiple line outage subsets. It has been shown that

Under multiple contingencies LMP-based congestion prices fluctuate considerably. Proposed method employs reduced problem formulation and decouples economic dispatch problem from state and LMP calculation problem. Thus, the large multiple contingency optimization problem can be solved efficiently. We believe that the proposed method will be very effective on networks of practical size. Based on Monte Carlo importance sampling idea, the proposed algorithm can stochastically assess the impact of multiple contingencies on LMP-congestion prices.

6.2 Future Work

Future work can be extended in following directions

- Explore possible ways of reducing computational effort in TR method by solving inner iterations using LSQR method
- Testing of the proposed LP based CCOPF with importance sampling

Appendix A

Network Test Cases

A.1 Introduction

Bus/branch network models are most commonly used in state estimation and power flow studies. The algorithms in this dissertation have been tested by means of a standard IEEE test systems that can be found in [90]. In power system state estimation the measurement set is usually a mixture of line power flow (both active and reactive), power bus injection (also active and reactive), and voltage magnitude measurement. Today even power angle measurements are available by means of PMUs, although those types of measurement were not consider in our study.

A fundamental question one has to answer when placing measurements is the following: "Is it possible to estimate the state from an available set of measurements, or in other words is the network observable?" An observability analysis is conducted prior to performing state estimation. Observability analysis is based on three methodologies: topological, numerical or hybrid. The topologically based algorithm that determines observability of the network was introduced by Clements and Wollenberg in [19] and further developed by Krumpholz, Clements and Davis in [52], where more details can be found. A review of the observability analysis methods and meter placement was prepared by Clements in [15].

A.2 IEEE 14 bus network case

The one-line diagram of the IEEE 14-bus network with a measurement set is illustrated in Fig. A.1. This network has been used in many examples throughout the research and also in many references cited in this dissertation. The original network and data files can be found in [90].

The IEEE 14-bus network in Fig. A.1 could be summarized:

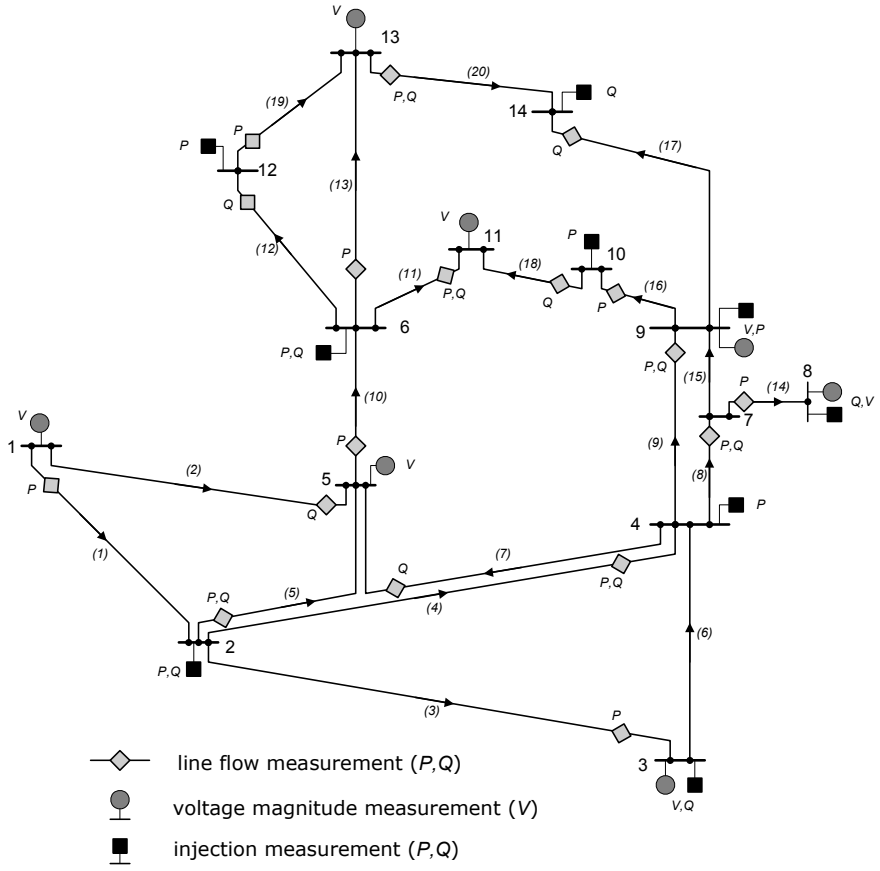

Figure A.1: IEEE 14-bus test system with measurement set

- number of buses: $N = 14$
- number of state variables: $n = 2N - 1 = 27$
- number of measurements: $m = 42$
- redundancy ratio $\eta = m/n = 1.56$

For practical implementation, there should be enough redundancy in measurement throughout the network. Degree of redundancy is usually expressed in terms of ratio of number of meters to number of states. η is a very important quantity, more redundant measurements give more chances for bad data to be detected [16].

Each of these measurements is not perfect. There is a constant level of error/noise present in the measurement. Therefore measurement error must be considered. The measurement error variance

σ^2, is assigned to each measurement type to reflect the expected accuracy of the meter used. These values are usually used as weights in the diagonal matrix R^{-1}. Assumed values of the variance σ^2 depending on the measurement type are given in Tables A.1 and A.2.

The way that we generated the measurement set is by calculating "perfect measurements" from the data available. Standard IEEE systems come with both parameters and solution. Measurement system is generated knowing the solution and then measurement noise (Gaussian random variable, zero mean unit variance) has been added to the perfect measurement to produce more realistic "noisy" measurements.

Table A.1: IEEE 14-bus test case - measurement set

type #	**measurement type**	**# of meas.**	σ^2
1	P flow	13	$1 \cdot 10^{-3}$
2	P injection	6	$1 \cdot 10^{-3}$
3	Q flow	11	$1 \cdot 10^{-3}$
4	Q injection	5	$1 \cdot 10^{-3}$
5	V magnitude	7	$1 \cdot 10^{-4}$

A.3 IEEE 30 bus network case

IEEE 30-bus network in Fig. A.2 could be summarized:

- number of buses: $N = 30$

- number of state variables: $n = 2N - 1 = 59$

- number of measurements: $m = 81$

- redundancy ratio $\eta = m/n = 1.37$

Table A.2: IEEE 30-bus test case - measurement set

type #	**measurement type**	**# of meas.**	σ^2
1	P flow	26	$1 \cdot 10^{-3}$
2	P injection	13	$1 \cdot 10^{-3}$
3	Q flow	26	$1 \cdot 10^{-3}$
4	Q injection	13	$1 \cdot 10^{-3}$
5	V magnitude	3	$1 \cdot 10^{-4}$

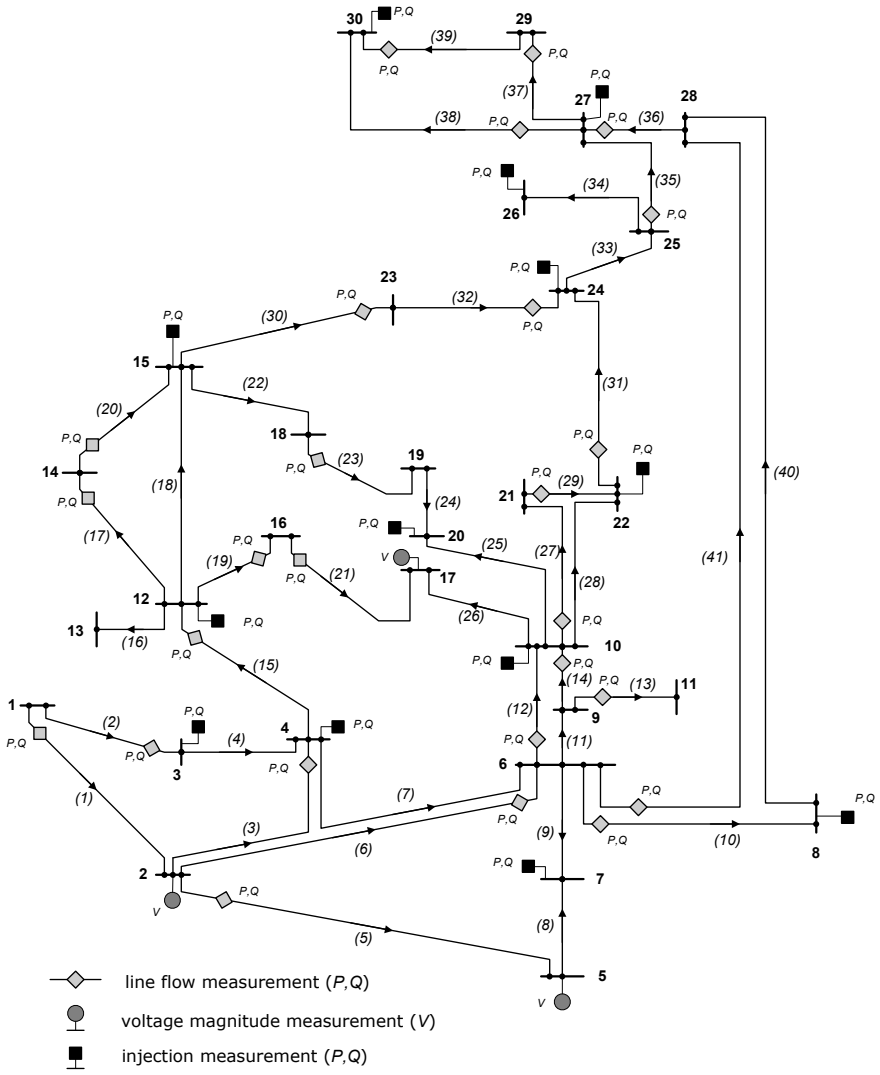

Figure A.2: IEEE 30-bus test system with measurement set

A.4 Non-converging cases

When we say "non-converging cases", we mean that the measurement set with topology error could not be solved by the Newton-QR algorithm. The notion of observability applied to the network with topology errors also. The design goal is to provide network observability under most operating conditions. If the outages or topology errors render a network unobservable even, the most robust algorithm won't be able to find the solution. While there is a constant effort to provide observable networks, temporary unobservability may still occur due to unanticipated network topology or failure in the telemetered measurements.

When building "non-converging" cases such as the ones in Fig. A.3 and Fig. A.4, we carefully placed the measurement set so that the network is observable. In Fig. A.3 and Fig. A.4 we denoted topology error by a dashed line, in which we assume that the line is out when it is actually in.

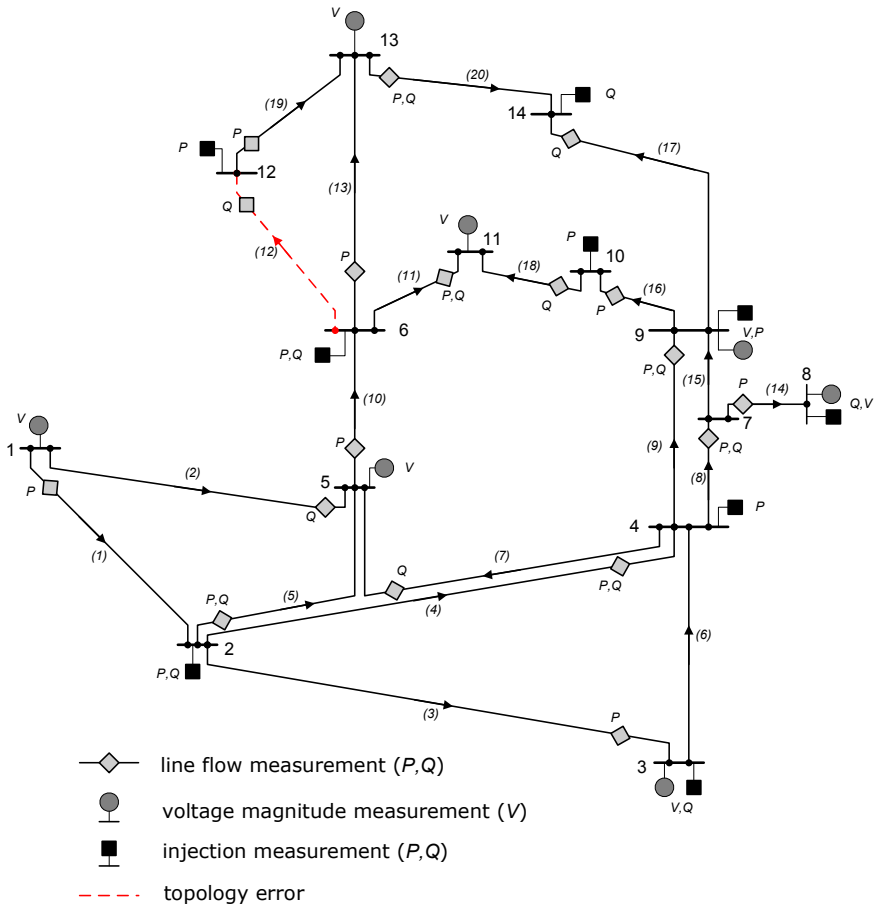

Figure A.3: IEEE 14-bus test system with measurement set and topology errors

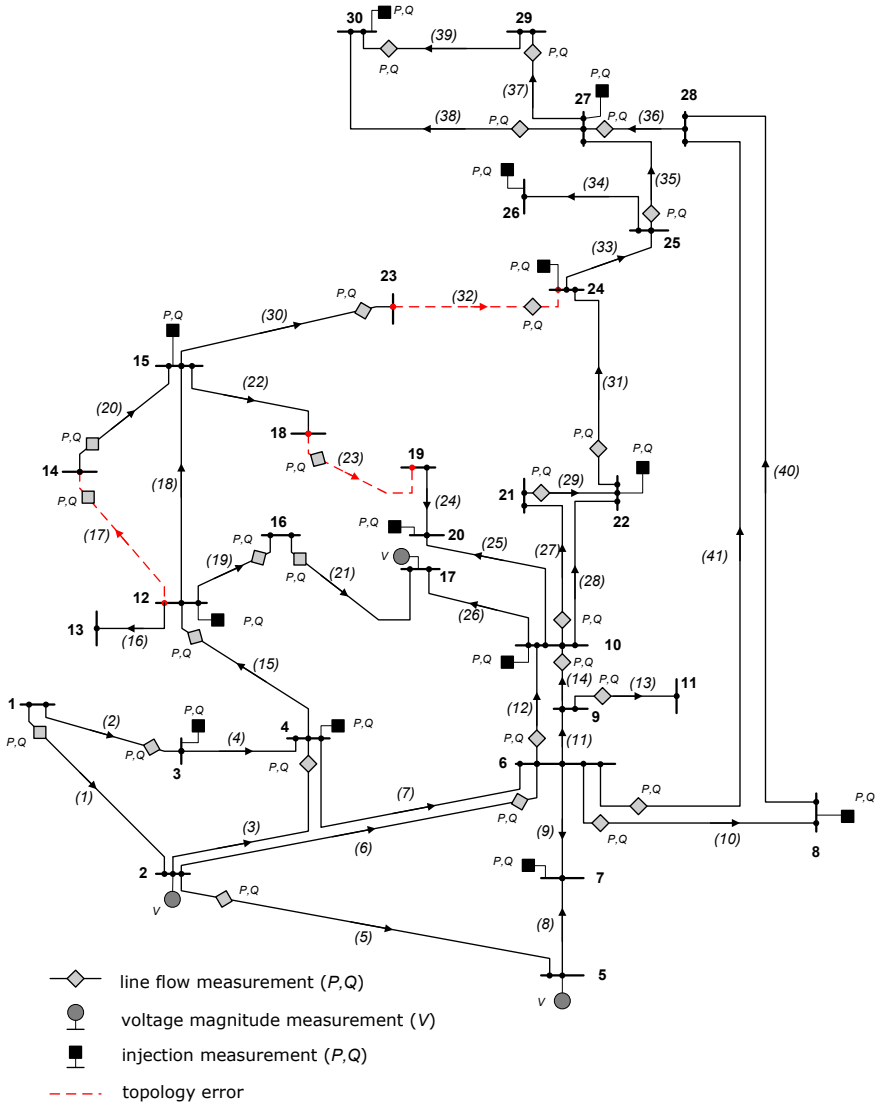

Figure A.4: IEEE 30-bus test system with measurement set and topology errors

Appendix B

B Matrix Theorems

In this Appendix we will prove four important theorems regarding the bus susceptance network matrix B and its modifications (i.e., matrices B' and $\widehat{B}$). Theorem B.4. is the key theorem in the development of the economic dispatch-based reduced system in Chapter 5. In order to prove Theorem B.4., Theorems B.1. through B.3. are needed.

Theorem B.1. is considered something of a Folk Theorem in the power system analysis community. To the best of the author's knowledge it has not been given a rigorous mathematical proof. Therefore, for completeness, we provide a mathematical proof for the fact that was taken for granted in many references.

Recall that $B \in \mathbb{R}^{n\times n}$ is a symmetric, singular matrix whose rows/columns have the following property

$$b_{kk} = -\sum_{\substack{j=1 \\ j\neq k}}^{n} b_{kj} \qquad k = 1, \ldots, n$$

Theorem B.1. *Suppose that matrix $B \in \mathbb{R}^{n\times n}$ is a symmetric matrix such that for, $k = 1, \ldots, n$*

$$b_{kk} < 0, \quad \textit{and} \quad b_{ik} \geq 0 \quad \textit{for} \quad i \neq k$$

and

$$b_{kk} = -\sum_{\substack{j=1 \\ j\neq k}}^{n} b_{kj} \qquad k = 1, \ldots, n$$

Then $\dim \mathcal{N}(B) = 1$*, where $\mathcal{N}(B)$ denotes the null-space of B.*

Proof. *Suppose that:*

$$B \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{pmatrix} = 0$$

Claim:

$$\begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{pmatrix} = \lambda \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \qquad \textit{for some } \lambda \in \mathbb{R}.$$

Suppose that not all v_i's have the same value. Then for some l, $1 \leq l \leq n$

$$|v_l| \geq |v_j| \qquad \textit{for} \quad 1 \leq j \leq n \quad \textit{and}$$
$$|v_l| > |v_j| \qquad \textit{for some} \quad k \neq l.$$

Then since

$$\sum_{j=1}^{n} b_{lj} v_j = 0$$

$$\begin{aligned} |b_{ll}||v_l| = |b_{ll} v_l| &= \Big| - \sum_{\substack{j=1 \\ j \neq l}}^{n} b_{lj} v_j \Big| \\ &\leq \sum_{\substack{j=1 \\ j \neq l}}^{n} |b_{lj}||v_j| \\ &< \Big(\sum_{\substack{j=1 \\ j \neq l}}^{n} |b_{lj}| \Big) |v_l| \\ &= \Big(\sum_{\substack{j=1 \\ j \neq l}}^{n} b_{lj} \Big) |v_l| \\ &= |b_{ll}||v_l| \end{aligned}$$

Which is a contradiction and therefore all v_i's must have the same value; hence $\dim \mathcal{N}(B) = 1$

Theorem B.2. *Suppose matrices B' and $\widehat{B}$ are defined as*

$$B' = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ b_{12} & b_{22} & \cdots & b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{1n} & b_{2n} & \cdots & b_{nn} \end{pmatrix} \qquad \text{and} \qquad \widehat{B} = \begin{pmatrix} b_{22} & \cdots & b_{2n} \\ \vdots & \ddots & \vdots \\ b_{2n} & \cdots & b_{nn} \end{pmatrix}$$

with the following property

$$b_{kk} = -\sum_{\substack{j=1 \\ j \neq k}}^{n} b_{kj} \qquad k = 2, \ldots, n$$

Then matrices B' and $\widehat{B}$ are nonsingular.

Proof. *Let us denote*

$$B' = \begin{pmatrix} e_1^T \\ b_2 \\ \vdots \\ b_n \end{pmatrix} \qquad e_1 = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \in \mathbb{R}^{n \times 1} \qquad e = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \in \mathbb{R}^{n \times 1}$$

It is straightforward to show that $\det(B') = \det(\widehat{B})$, so $\widehat{B}$ is nonsingular if and only if B' is nonsingular.

Also due to the property of the B matrix

$$Bv = 0 \qquad \Leftrightarrow \qquad v = \lambda \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$$

Since $\dim \mathcal{N}(B) = 1$, where $\mathcal{N}(B)$ denotes the null-space of B, $b_2, \ldots, b_n$ of B are linearly independent. In order to prove that, suppose a contradiction.

Assume that vectors $b_2, \ldots, b_n$ are linearly dependent vectors. Then

$$\sum_{i=2}^{n} \alpha_i b_i = 0$$

for some α_i's that are not all zero. Then

$$\sum_{i=2}^{n} \alpha_i b_i = 0 \qquad \Rightarrow \qquad B \begin{pmatrix} 0 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{pmatrix} = 0 \qquad \Rightarrow \qquad \dim \mathcal{N}(B) \geq 2$$

which is a contradiction.

Now suppose $B'v = 0$ for some v. Then

$$0 = B'v = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ b_{12} & b_{22} & \cdots & b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{1n} & b_{2n} & \cdots & b_{nn} \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{pmatrix} = \begin{pmatrix} v_1 \\ b_2 v \\ \vdots \\ b_n v \end{pmatrix}$$

We have

$$0 = b_2 v = \ldots = b_n v$$

Therefore, v is orthogonal to the linearly independent rows $b_2, \ldots, b_n$ of B i.e.,

$$v \in span\{b_2^T, \ldots, b_n^T\} \perp \{\lambda e : \lambda \in \mathbb{R}\}$$

$$\Rightarrow \qquad v = \lambda \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$$

But $v_1 = \lambda = 0 \Rightarrow \lambda = 0$, and $v = 0$. Therefore $B'v = 0$ only if $v = 0$; thus B' and $\widehat{B}$ are nonsingular.

Theorem B.3. *Given:*

$$B' = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ b_{12} & b_{22} & \cdots & b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{1n} & b_{2n} & \cdots & b_{nn} \end{pmatrix} = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ b_{12} & & & \\ \vdots & & \widehat{B} & \\ b_{1n} & & & \end{pmatrix}$$

with the property

$$b_{kk} = -\sum_{\substack{j=1 \\ j \neq k}}^{n} b_{kj} \qquad k = 2, \ldots, n$$

then

$$B'^{-1} = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 1 & & & \\ \vdots & & \widehat{B}^{-1} & \\ 1 & & & \end{pmatrix}$$

Proof. *Set:*

$$C = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 1 & & & \\ \vdots & & \widehat{B}^{-1} & \\ 1 & & & \end{pmatrix}$$

Then the first column of $B'C$ is

$$\begin{pmatrix} 1 & 0 & \cdots & 0 \\ b_{12} & & & \\ \vdots & & \widehat{B}^{-1} & \\ b_{1n} & & & \end{pmatrix} \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 1 & & & \\ \vdots & & \widehat{B}^{-1} & \\ 1 & & & \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

and the second through nth columns are:

$$B' \begin{pmatrix} 0 & \cdots & 0 \\ & & \\ & \widehat{B} & \\ & & \end{pmatrix} = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ b_{21} & & & \\ \vdots & & \widehat{B} & \\ b_{n1} & & & \end{pmatrix} \begin{pmatrix} 0 & \cdots & 0 \\ & & \\ & \widehat{B}^{-1} & \\ & & \end{pmatrix}$$

$$= \begin{pmatrix} 0 & \cdots & 0 \\ & & \\ & \widehat{B}\widehat{B}^{-1} & \\ & & \end{pmatrix} = \begin{pmatrix} 0 & \cdots & 0 \\ 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & & 1 \end{pmatrix}$$

It follows than that $B'C = I$, so $C = B'^{-1}$

Theorem B.4. *Suppose B, $(B = B^T)$ and B' are defined as*

$$B = \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{12} & b_{22} & \cdots & b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{1n} & b_{2n} & \cdots & b_{nn} \end{pmatrix} \qquad \textit{and} \qquad B' = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ b_{12} & b_{22} & \cdots & b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{1n} & b_{2n} & \cdots & b_{nn} \end{pmatrix}$$

with the property

$$b_{kk} = -\sum_{\substack{j=1 \\ j \neq k}}^{n} b_{kj} \qquad k = 1, \ldots, n$$

Then

$$B \cdot B'^{-1} = \begin{pmatrix} 0 & -1 & \cdots & -1 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix}$$

Proof. *Let $D = B \cdot B'^{-1}$. We claim that*

$$D = B \cdot B'^{-1} = \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{12} & & & \\ \vdots & & \widehat{B} & \\ b_{1n} & & & \end{pmatrix} \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 1 & & & \\ \vdots & & \widehat{B}^{-1} & \\ 1 & & & \end{pmatrix} = \begin{pmatrix} 0 & -1 & \cdots & -1 \\ 0 & & & \\ \vdots & & \widehat{B}\widehat{B}^{-1} & \\ 0 & & & \end{pmatrix}$$

Since $b_{11} = -\sum_{i=1}^{n} b_{1n}$, it is straightforward to show that the first column of matrix D is the zero vector. We have to show that

$$D_{1j} = -1 \qquad for \quad j = 2, \ldots, n$$

Recall that if we multiply matrices $P \in \mathbb{R}^{m \times p}$ and $Q \in \mathbb{R}^{p \times n}$, then the product $W \in \mathbb{R}^{m \times n}$ is

$$W_{ij} = \sum_{k=1}^{n} P_{ik} Q_{kj}$$

or if p_i is the ith row vector of matrix P and vector q_j is the jth column vector of matrix Q, then the matrix product can be written

$$W_{ij} = p_i^T q_j$$

Accordingly, if we define the elements of matrix $\widehat{B}$ as $\widehat{b}_{ij}$ and the elements of matrix $\widehat{B}^{-1}$ as $\widetilde{b}_{ij}$, then the first row elements of matrix D are

$$D_{1j} = \sum_{k=2}^{n} b_{1k} \widetilde{b}_{kj} \qquad j = 2, \ldots, n \tag{B.1}$$

Using the given property of the row/column elements of matrix B

$$b_{1k} = -\sum_{i=2}^{n} \widehat{b}_{ik} \qquad for \quad k = 2, \ldots, n$$

then equation (B.1) can be rewritten as

$$D_{1j} = -\sum_{k=2}^{n} \sum_{i=2}^{n} \widehat{b}_{ik} \widetilde{b}_{kj}$$

If we denote by $\widehat{b}_i$ *the ith row of* $\widehat{B}$ *and by* $\widetilde{b}_j$ *the jth column of* $\widehat{B}^{-1}$*, then*

$$D_{1j} = -\sum_{i=2}^{n} \widehat{b}_i^T \widetilde{b}_j$$

or, in other words, D *is a negative sum of dot products of all rows of* $\widehat{B}$ *with the jth column of* $\widehat{B}^{-1}$*. One can see that only the jth element of the sum produces a nonzero element; moreover* $\widehat{b}_j^T \widetilde{b}_j = 1$*. Hence,*

$$D_{1j} = -1 \qquad for \quad j = 2, \ldots, n$$

Bibliography

[1] A. Abur and A. G. Expósito, *Power System State Estimation Theory and Implementation*. New York: Marcel Dekker, 2004.

[2] J. Allemong, "State estimation fundamentals for successful deployment," in *Proc. IEEE PES General Meeting*, San Francisco, CA, June 2005.

[3] O. Alsaç, J. Bright, M. Prais, and B. Stott, "Further development in LP-based optimal power flow," *IEEE Trans. Power Syst.*, vol. 5, no. 3, pp. 697–711, Aug. 1990.

[4] O. Alsaç, J. M. Bright, S. Brignone, M. Prais, C. Silva, B. Stott, and N. Vempati, "The right to fight price volatility," *IEEE Power Energy Mag.*, vol. 2, no. 4, pp. 47–57, July/August 2004.

[5] O. Alsaç and B. Stott, "Optimal load flow with steady-state security," *IEEE Trans. Power App. Syst.*, vol. PAS-93, pp. 745–751, May/June 1974.

[6] O. Alsaç, N. Vempati, B. Sttot, and A. Monticelli, "Generalized state estimation," *IEEE Trans. Power Syst.*, vol. 13, no. 3, pp. 1069–1075, Aug. 1998.

[7] F. L. Alvarado, W. F. Tinney, and M. K. Enns, "Sparsity in large-scale network computation," in *Control and Dynamic Systems, Advances in Theory and Applications*, ser. Analysis and Control System Techniques for Electric Power Systems Part 1 of 4, C. T. Leondes, Ed. San Diego, CA: Academic Press, 1991, vol. 41, pp. 207–272.

[8] F. C. Aschmoneit, N. M. Peterson, and E. C. Adrian, "State estimation with equality constraints," in *10th PICA Conf. Proc.*, Toronto, Canada, May 1977, pp. 427–430.

[9] R. Barrett, M. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo,

C. Romine, and H. V. der Vorst, *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*, 2nd ed. Philadelphia, PA: SIAM, 1994.

[10] D. P. Bertsekas, *Nonlinear Programming.* Belmont, MA: Athena Scientific, 1999.

[11] A. Bose and K. A. Clements, "Real-time modeling of power networks," *Proc. IEEE*, vol. 75, no. 12, pp. 1607–1622, Dec. 1987.

[12] R. C. Burchett, H. H. Happ, and D. R. Vierath, "Quadratically convergent optimal power flow," *IEEE Trans. Power App. Syst.*, vol. PAS-103, no. 11, pp. 3267–3275, Nov. 1984.

[13] M. B. Cain and F. L. Alvarado, "Implications of cost and bid format on electricity market studies: Linear versus quadratic costs," in *Large Engineering Systems Conference on Power Engineering*, Halifax, Canada, July 2004.

[14] J. Carpentier, "Contribution a l'etude du dispatching economique," in *Bull. Soc. Francaise Electriciens*, vol. 3, Aug 1962.

[15] K. A. Clements, "Observability methods and optimal meter placement," *Int. J. Elec. Power and Energy*, vol. 12, no. 2, pp. 89–93, Apr. 1990.

[16] K. A. Clements and P. W. Davis, "Detection and identification of topology errors in electric power systems," *IEEE Trans. Power Syst.*, vol. 3, no. 4, pp. 1748–1753, Nov. 1988.

[17] K. A. Clements, P. W. Davis, and K. D. Frey, "An interior point algorithm for weighted least absolute value power state estimation," in *Proc. IEEE/PES Winter Meeting*, 1991, paper 91 WM 235-2 PWRS.

[18] ——, "Treatment of inequality constraints in power system state estimation," *IEEE Trans. Power Syst.*, vol. 10, no. 2, pp. 567–574, May 1995.

[19] K. A. Clements and B. F. Wollenberg, "An algorithm for observability determination in power system state estimation," in *Proc. IEEE Summer Power Meeting*, 1975, paper A 75 447-3.

[20] A. R. Conn, N. I. M. Gould, and P. L. Toint, *Trust-Region Methods.* Philadelphia, PA: SIAM, 2000.

[21] M. B. Coutto, A. M. L. Silva, and D. M. Falcão, "Bibliography on power system state estimation (1968-1989)," *IEEE Trans. Power Syst.*, vol. 5, no. 3, pp. 950–961, Aug. 1990.

[22] H. Dağ and F. L. Alvarado, "Toward improved uses of the conjugate gradient method for power system applications," *IEEE Trans. Power Syst.*, vol. 12, no. 3, pp. 1306–1314, Aug. 1997.

[23] H. Dağ and A. Semlyen, "A new preconditioned conjugate gradient power flow," *IEEE Trans. Power Syst.*, vol. 18, no. 4, pp. 1248–1255, Nov. 2003.

[24] J. E. Dennis and R. E. Schnabel, *Numerical Methods for Unconstrained Minimization and Nonlinear Equations*, 2nd ed., ser. Classics in Applied Mathematics. Philadelphia, PA: SIAM, 1996.

[25] H. W. Dommel and W. F. Tinney, "Optimal power flow solutions," *IEEE Trans. Power App. Syst.*, vol. PAS-87, pp. 1866–1876, Oct. 1968.

[26] J. Doudna and D. Salem-Natarajan, "State estimation issues facing ISO/RTO organizations," in *Proc. IEEE PES General Meeting*, San Francisco, CA, June 2005.

[27] T. E. Dy Liacco, "The role and implementation of state estimation in an energy management system," *Int. J. Elec. Power and Energy*, vol. 12, no. 2, pp. 75–79, Apr. 1990.

[28] R. Ebrahimian and R. Baldick, "State estimator condition number analysis," *IEEE Trans. Power Syst.*, vol. 16, no. 2, pp. 273–279, May 2001.

[29] H. Elman, "A stability analysis of incomplete LU factorization," *Math. Comp.*, vol. 47, pp. 191–218, 1986.

[30] R. Entriken and G. Infanger, "Decomposition and importance sampling for stochastic linear models," *Energy, The International Journal*, vol. 15, no. 7/8, pp. 645–659, July-August 1990.

[31] F. D. Galiana, H. Javidi, and S. McFee, "On the application of a pre-conditioned conjugate gradient algorithm to power network analysis," *IEEE Trans. Power Syst.*, vol. 9, no. 2, pp. 629–636, May 1994.

[32] A. Gjelsvik, S. Aam, and L. Holten, "Hachtel's augmented matrix method - a rapid method improving numerical stability in power system static state estimation," *IEEE Trans. Power App. Syst.*, vol. PAS-104, no. 11, pp. 2987–2993, November 1985.

[33] H. Glavitsch and R. Bacher, "Optimal power flow algorithms," in *Control and Dynamic Systems, Advances in Theory and Applications*, ser. Analysis and Control System Techniques

for Electric Power Systems Part 1 of 4, C. T. Leondes, Ed. San Diego, CA: Academic Press, 1991, vol. 41, pp. 135–204.

[34] G. Golub and W. Kahan, "Calculating the singular values and pseudo-inverse of a matrix," *SIAM J. Numer. Anal.*, vol. 2, no. 2, pp. 205–224, 1965.

[35] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed. The Johns Hopkins University Press, 1996.

[36] J. W. Gu, K. A. Clements, G. R. Krumpholz, and P. W. Davis, "The solution of ill-conditioned power system state estimation problems via the method of Peter and Wilkinson," *IEEE Trans. Power App. Syst.*, vol. 102, no. 10, pp. 3473–3480, Oct. 1983.

[37] J. M. Hammersley and D. C. Handscomb, *Monte Carlo Methods*. London, UK: Methuen & Co Ltd, 1964.

[38] M. Hestenes and E. Stiefel, "Methods of conjugate gradients for solving linear systems," *J. Res. National Bureau of Standards*, vol. 49, pp. 409–439, 1952.

[39] W. W. Hogan, "Contract networks for electric power transmission: Technical reference," John F. Kenedy School of Government, Harvard University, Cambridge, MA, Tech. Rep., February 1992.

[40] L. Holten, A. Gjelsvik, S. Aam, F. F. Wu, and W. H. E. Liu, "Comparison of different methods for state estimation," *IEEE Trans. Power Syst.*, vol. 3, no. 4, pp. 1798–1806, Nov. 1988.

[41] M. Huneault and F. D. Galiana, "A survey of the optimal power flow literature," *IEEE Trans. Power Syst.*, vol. 6, no. 2, pp. 762–770, May 1991.

[42] M. Ilić, "Transmission reliability and security under open access," in *Proc. IEEE PES General Meeting*, Denver, CO, June 2004, Invited Panel.

[43] M. Ilić, F. Galiana, and L. Fink, *Power Systems Restructuring: Engineering and Economics*. Kluwer Academic Publisher, 1998.

[44] M. D. Ilić, E. H. Allen, J. W. Chapman, C. A. King, J. H. Lang, and E. Litvinov, "Preventing future blackouts by means of enhanced power system control: From complexity to order," *Proc. IEEE*, vol. 93, no. 11, pp. 1920–1941, Nov. 2005.

[45] M. Ilić-Spong and A. Phadke, "Redistribution of reactive power flow in contingency studies," *IEEE Trans. Power Syst.*, vol. 1, no. 3, pp. 266–275, Aug. 1996.

[46] G. Infanger, *Planning Under Uncertainty Solving Large-Scale Stochastic Linear Problems.* Danvers, MA: Boyd & Fraser Publishing Company, 1994.

[47] G. Irisarri, L. M. Kimball, K. A. Clements, A. Bagchi, and P. W. Davis, "Economic dispatch with network and ramping constraints via interior point methods," *IEEE Trans. Power Syst.*, vol. 13, no. 1, pp. 236–242, Feb. 1998.

[48] C. T. Kelley, *Solving Nonlinear Equations with Newton's Method*, ser. Fundamentals of Algorithms. Philadelphia, PA: SIAM, 2003.

[49] L. M. Kimball, K. A. Clements, and P. W. Davis, "Stochastic OPF via Bender's method," in *IEEE Power Tech Conf.*, Porto, Portugal, Sep 2001.

[50] ——, "An implementation of the stochastic OPF problem," *Elect. Power Compon. Syst*, vol. 31, pp. 1193–1204, Dec. 2003.

[51] L. M. Kimball, K. A. Clements, S. Pajić, and P. W. Davis, "Stochastic OPF via constraint relaxation," in *IEEE Power Tech Conf.*, Bologna, Italy, June 2003.

[52] G. R. Krumpholz, K. A. Clements, and P. W. Davis, "Power system observability: A practical algorithm using network topology," *IEEE Trans. Power App. Syst.*, vol. PAS-99, no. 4, pp. 1534–1542, July/Aug 1980.

[53] K. Levenberg, "A method for the solution of certain nonlinear problems in least squares," *Quart. J. Appl. Math.*, vol. 2, pp. 164–168, 1944.

[54] D. W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *SIAM J. Appl. Math.*, vol. 11, no. 2, pp. 431–441, Jun 1963.

[55] A. Marshall, "The use of multi-stage sampling schemes in Monte Carlo computations," in *Symposium of Monte Carlo Methods*, M. Meyer, Ed. New York: Wiley, 1956, pp. 123–140.

[56] T. A. Mikolinnas and B. F. Wollenberg, "An advanced contingency selection algorithm," *IEEE Trans. Power App. Syst.*, vol. PAS-100, no. 2, pp. 608–617, Feb. 1981.

[57] A. Monticelli, *State Estimation in Electric Power Systems - A General Approach.* Boston: Kluwer Academic Publishers, 1999.

[58] A. Monticelli, M. V. F. Pereira, and S. Granville, "Security-constrained optimal power flow with post-contingency corrective rescheduling," *IEEE Trans. Power Syst.*, vol. 2, no. 1, pp. 175–182, Feb. 1987.

[59] J. J. Moré, "The Levenberg-Marquardt algorithm: Implementation and theory," in *Numerical Analysis, Dundee 1977*, ser. Lecture Notes in Mathematics, G. Watson, Ed. New York, NY: Springer-Verlag, 1977, vol. 630, pp. 105–116.

[60] J. J. Moré and D. C. Sorensen, "Computing a trust region step," *SIAM J. Sci. Statist. Comput.*, vol. 3, no. 4, pp. 553–572, Sep 1983.

[61] ——, "Newton's method," in *Studies in Numerical Analysis*, ser. MAA Studies in Mathematics, G. Golub, Ed. Providence, RI: American Mathematical Society, 1984, vol. 24, pp. 29–81.

[62] I. M. Nejdawi, "Optimal power flow using sequential quadratic programming," Ph.D. dissertation, Worcester Polytechnic Institute, Worcester, MA, 1999.

[63] I. M. Nejdawi, K. A. Clements, and P. W. Davis, "An efficient interior point method for sequential quadratic programming based optimal power flow," *IEEE Trans. Power Syst.*, vol. 15, no. 4, pp. 1179–1183, Nov. 2000.

[64] J. Nieplocha and C. C. Carroll, "Iterative methods for the WLS state estimation on RISC, vector, and parallel computers," in *Proceedings of the North American Power Symposium*, Washington, DC, Oct. 1993, pp. 355–363.

[65] J. Nieplocha, A. Marquez, V. Tipparaju, D. Chavarria-Miranda, R. Guttromson, and H. Huang, "Towards efficient power system state estimators on shared memory computers," in *Proceedings of the IEEE PES General Meeting*, Montreal, CA, June 2006.

[66] T. J. Overbye, X. Cheng, and Y. Sun, "A comparison for AC and DC power flow models for LMP calculations," in *Proceedings of the 37th Hawaii International Conference on System Science*, 2004.

[67] C. C. Paige and M. A. Saunders, "Algorithm 583 LSQR: Sparse linear equation and sparse least squares problems," *ACM Trans. Math. Soft.*, vol. 8, pp. 195–209, 1982.

[68] ——, "LSQR: An Algorithm for Sparse Linear Equation and Sparse Least Squares," *ACM Trans. Math. Soft.*, vol. 8, pp. 43–71, 1982.

[69] S. Pajić, "Sequential quadratic programming-based contingency constrained optimal power flow," Master's thesis, Worcester Polytechnic Institute, Worcester, MA, 2003.

[70] S. Pajić and K. A. Clements, "Globally convergent state estimation via the trust region method," in *Proc. IEEE Power Tech Conf.*, Bologna, Italy, June 23-26 2003.

[71] ——, "Power system state estimation via globally convergent methods," *IEEE Trans. Power Syst.*, vol. 20, no. 4, pp. 1683–1689, Nov. 2005.

[72] M. Rice and G. T. Heydt, "Phasor measurement unit data in power system state estimation," PSERC," Intermediate Project Report for PSERC Project "Enhanced State Estimators", 2005. [Online]. Available: http://www.pserc.org

[73] Y. Saad, *Iterative Methods for Sparse Linear Systems.* Boston: PWS Publishing Co., 1996.

[74] F. C. Schweppe, "Power system static state estimation, part III: Implementation," *IEEE Trans. Power App. Syst.*, vol. PAS-89, no. 2, pp. 130–135, Jan. 1970.

[75] F. C. Schweppe, M. C. Caramanis, R. D. Tabors, and R. E. Bohn, *Spot Pricing of Electricity.* Kluwer Academic Publisher, 1988.

[76] F. C. Schweppe and D. B. Rom, "Power system static state estimation, part II: Approximate model," *IEEE Trans. Power App. Syst.*, vol. PAS-89, no. 2, pp. 125–130, Jan. 1970.

[77] F. C. Schweppe and J. Wilders, "Power system static state estimation, part I: Exact model," *IEEE Trans. Power App. Syst.*, vol. PAS-89, no. 2, pp. 120–125, Jan. 1970.

[78] A. Semlyen, "Fundamental concepts of a Krylov subspace power flow methodology," *IEEE Trans. Power Syst.*, vol. 11, no. 3, pp. 1528–1537, Aug. 1996.

[79] G. A. Shultz, R. B. Schnabel, and R. H. Byrd, "A family of trust region algorithms for unconstrained minimization with strong global convergence properties," *SIAM J. Numer. Anal.*, vol. 22, no. 1, pp. 47–67, Feb 1985.

[80] A. Simões-Costa and V. H. Quintana, "An orthogonal row processing algorithm for power system sequential state estimation," *IEEE Trans. Power App. Syst.*, vol. PAS-100, no. 8, pp. 3791–3800, August 1981.

[81] ——, "A robust numerical technique for power system state estimation," *IEEE Trans. Power App. Syst.*, vol. PAS-100, no. 2, pp. 691–698, February 1981.

[82] I. M. Sobol, *A Primer for the Monte Carlo Method*. CRC Press Inc., 1994.

[83] D. C. Sorensen, "Newton's method with a model trust region modification," *SIAM J. Numer. Anal.*, vol. 19, no. 2, pp. 409–426, Apr 1982.

[84] B. Stott and O. Alsaç, "Fast decoupled load flow," *IEEE Trans. Power App. Syst.*, vol. PAS-93, pp. 859–869, May/June 1974.

[85] B. Stott, O. Alsaç, and A. J. Monticelli, "Security analysis and optimization," *Proc. IEEE*, vol. 75, no. 12, pp. 1623–1644, Dec. 1987.

[86] B. Stott and E. Hobson, "Power system security control calculations using linear programming part I and II," *IEEE Trans. Power App. Syst.*, vol. 97, no. 5, pp. 1713–1731, Sept/Oct 1978.

[87] B. Stott and J. L. Marinho, "Linear programming for power-system network security applications," *IEEE Trans. Power App. Syst.*, vol. 98, no. 3, pp. 837–848, May/June 1979.

[88] D. I. Sun, B. Ashley, B. Brewer, A. Hughes, and W. F. Tinney, "Optimal power flow by Newton approach," *IEEE Trans. Power App. Syst.*, vol. PAS-103, no. 10, pp. 2864–2880, Oct. 1984.

[89] W. F. Tinney and C. E. Hart, "Power flow solution by Newton's method," *IEEE Trans. Power App. Syst.*, vol. PAS-86, no. 11, pp. 1447–1460, Nov. 1967.

[90] Data for the IEEE 14, 30, 57, 118, 300-bus test system. University of Washington. [Online]. Available: http://www.ee.washington.edu/research/pstca/

[91] Final Report on the August 14, 2003, Blackout in the United States and Canada: Causes and Recommendations. U.S.-Canada Power System Outage Task Force. [Online]. Available: https://reports.energy.gov/

[92] R. A. M. Van Amerongen, "On convergence analysis and convergence enhancement of power system least-squares state estimation," *IEEE Trans. Power Syst.*, vol. 10, no. 4, pp. 2038–2044, Nov. 1995.

[93] N. Vempati, I. W. Slutsker, and W. Tinney, "Enhancement to Givens rotations for power system state estimation," *IEEE Trans. Power Syst.*, vol. 6, no. 4, pp. 842–849, Nov. 1991.

[94] H. F. Walker, "Numerical methods for nonlinear equations," WPI Mathematical Sciences Department, Tech. Rep. MS-03-02-18, 2002.

[95] ——, "Lecture notes: Numerical linear algebra," 2005, WPI Mathematical Sciences Department.

[96] ——, "Lecture notes: Numerical methods for nonlinear equations and unconstrained optimization," 2006, WPI Mathematical Sciences Department.

[97] A. J. Wood and B. F. Wollenberg, *Power Generation Operation and Control*, 2nd ed. New York, NY: John Wiley & Sons, 1996.

[98] S. J. Wright, *Primal-Dual Interior-Point Methods*. Philadelphia, PA: SIAM, 1997.

[99] F. Wu, P. Varaiya, P. Spiller, and S. Oren, "Folk theorems on transmission access: proofs and counterexamples," *Journal of Regulatory Economics*, vol. 10, no. 1, pp. 5–23, 1996.

[100] F. F. Wu, "Power system state estimation: a survey," *Int. J. Elec. Power and Energy*, vol. 12, no. 2, pp. 80–87, Apr. 1990.