AN ABSTRACT OF THE THESIS OF

Ruqi Li for the degree of Master of Science in

Electrical & Computer Engineering Presented on June 5, 1987

Title: APPLICATIONS OF MATHEMATICAL PROGRAMMING TECHNIQUES IN OPTIMAL POWER FLOW PROBLEMS

Abstract approved: Redacted for privacy

In the past two decades, a great amount of attention has been paid to investigating the economic and safe operation of power systems using modern mathematical techniques. The classic economic dispatch problem, now often called the Optimal Power Flow problem, has been formulated as a mathematical optimization problem and has been solved using various kinds of mathematical programming techniques, such as nonlinear, quadratic, linear and dynamic programming. This paper presents some of these achievements. The nonlinear and quadratic programming techniques employed to solve the Optimal Power Flow problem are mainly studied. The use of the linear programming technique is considered. Using these new approaches, the application of the Optimal Power Flow is not restricted only to economical
purposes for on-line real and reactive power control and voltage control but has been extended to include security considerations of power systems. Excellent work has been done to find Optimal Power Flow solutions. Various kinds of Optimal Power Flow programs have been commercially available. More work remains to be done to seek more reliable, faster solution algorithms.
APPLICATIONS OF MATHEMATICAL PROGRAMMING
TECHNIQUES IN OPTIMAL POWER FLOW PROBLEMS

by

Ruqi Li

A THESIS
submitted to
Oregon State University

in partial fulfillment of
the requirements for the
degree of
Master of Science

Completed  June 5, 1987
Commencement  June 1988
APPROVED:

Redacted for privacy

Associate Professor of Electrical & Computer Engr. in charge of major

Redacted for privacy

Chairman of the Department of Electrical & Computer Engineering

Redacted for privacy

Dean of the Graduate School

Date thesis is presented ___________ June 5, 1987

Typed by Lei Luo for ___________ Ruqi Li
ACKNOWLEDGEMENT

I am grateful to Dr. G.C. Alexander, my major professor and thesis advisor, for his overall guidance, many helpful suggestions and criticisms throughout my graduate study. His generous advice, assistance and friendship are greatly appreciated.

I wish to thank Dr. J.L. Saugen, Dr. H.K. Lauw and Dr. J.D. Ingle for their serving on my graduate committee and giving me a lot of valuable advice.

I would like to express my deep appreciation and respect to Professor J.F. Engle for his valuable assistance and guidance.

I thank all the teachers who have taught me during my graduate study.

I also would like to extend my sincere appreciation to my wife Lei Luo, my parents for their love, help, self-sacrifice and encouragement.
# TABLE OF CONTENTS

| SECTION 1 | INTRODUCTION | 1 |
| SECTION 2 | THE NATURE OF THE OPTIMAL POWER FLOW PROBLEM | 7 |
| SECTION 3 | FORMULATION OF THE OPTIMAL POWER FLOW PROBLEM | 12 |
| SECTION 4 | THE OPTIMAL POWER FLOW SOLUTIONS | 18 |
| 4.1 | The Optimal Power Flow Solutions Using Nonlinear Programming Methods | 18 |
| 4.1.1 | The Optimal Power Flow Solutions Without the Inequality Constraints | 19 |
| 4.1.2 | The Optimal Power Flow Solution Involving the Inequality Constraints | 32 |
| 4.1.2.1 | Solution of (b) by the Penalty Function Method | 33 |
| 4.1.2.2 | Solution of (b) Using the Indirect Method | 42 |
| 4.2 | Optimal Power Flow Based on P-Q Decomposition | 45 |
| 4.3 | Contingency-Constrained Optimal Power Flow | 50 |
| 4.4 | The Optimal Power Flow Solutions Using Quadratic Programming | 56 |
| SECTION 5 | A SUMMARY OF THE CHARACTERISTICS OF THE VARIOUS OF METHODS | 62 |
| SECTION 6 | CONCLUSIONS/IMPLEMENTATION OF OPTIMAL POWER FLOW AT OSU | 66 |
| BIBLIOGRAPHY | 68 |
| APPENDICES | 71 |
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. A simplified flow chart of the algorithm</td>
<td>23</td>
</tr>
<tr>
<td>2. OPF algorithm involving inequality constraints</td>
<td>41</td>
</tr>
<tr>
<td>3. P-Q Decomposition Optimal Power Flow algorithm flow chart</td>
<td>49</td>
</tr>
</tbody>
</table>
LIST OF IMPORTANT NOTATIONS

\[ N \] — Number of buses of a power system.

\[ N_L \] — Number of P,Q buses.

\[ N_g \] — Number of P,V buses.

\[ P_{Lj} \] — Real power load at the jth bus.

\[ P_{Gj}=P_j \] — Real power output of the jth generator.

\[ Q_{Lj} \] — Reactive power load at the jth bus.

\[ Q_{Gj}=Q_j \] — Reactive power output of the jth generator.

\[ P_{TL} \] — Total transmission losses of the system.

\[ T_k \] — Tap setting of the kth tap-changing transformer.

\[ f(X,U) \] — Cost function.

\[ P_N \] — Real power output at the reference bus.

\[ P_{\text{IN}j} \] — Real power injected into the jth bus.

\[ Q_{\text{IN}j} \] — Reactive power injected into the jth bus.

\[ [X] \] — State vector.

\[ [U] \] — Control subvector (also called control vector).

\[ [G(X,U)]=[0] \] — Equality constraint vector function.

\[ [H(X,U)]\leq[0] \] — Inequality constraint vector function.

\[ [\lambda] \] — Lagrangian multiplier vector.

\[ L(X,U,\lambda) \] — Lagrange equation, or augmented objective function.

\[ [\ ]^t \] — Transpose of a vector.

\[ [X]^0, [U]^0 \] — Initial state, control vectors.

\[ [X]^k, [U]^k \] — State, control vectors at the kth iteration.

\[ \epsilon \] — Nonspecific convergence tolerance.
\[ \delta U^k \] — Changes in control vector.

\[ B \] — Gradient step parameter.

\[ \text{grad } f = [g] \] — Gradient vector of the cost function.

\[ \frac{\delta^2 f}{\partial u_i \partial u_j} = [A] \] — The Hessian matrix.

\[ [H] \] — Approximated Hessian matrix.

\[ [P] \] — Search direction vector.

\[ ||[g]|| \] — Norm of vector \([g]\).

\[ c,b \] — Penalty weighting parameters.

\[ w_j \] — Penalty function associated with the \(j\)th inequality constraint of \([X]\).

\[ x_j^{\text{max}}, x_j^{\text{min}} \] — Upper or lower limit of the \(j\)th state variable.

\[ u_j^{\text{max}}, u_j^{\text{min}} \] — Upper or lower limit of the \(j\)th control variable.

\[ [W(X,U)] \] — Penalty function vector.

\[ [U^P], [X^P] \] — Control, state vectors for P-0 problem.

\[ [U^Q], [X^Q] \] — Control, state vectors for Q-V problem.

\[ [U_k], [X_k] \] — Control, state vectors associated with the \(k\)th contingency case.

\[ [B] \] — B-coefficient matrix, penalty weighting parameter matrix (diagonal matrix).
APPLICATIONS OF MATHEMATICAL PROGRAMMING TECHNIQUES
IN OPTIMAL POWER FLOW PROBLEMS

SECTION I. INTRODUCTION

The development of economic dispatch, the predecessor of Optimal Power Flow, or OPF, dates back to the early 1920's, or even earlier, when two or more generation units were committed to take on loads on a power system whose total demand exceeded the generation capabilities of each generator. The problem that was encountered at that time was how to exactly divide up real power between two units such that the load was served and the power generation cost was minimized.

By the early 1930's, the equal incremental method had been developed, which has yielded the most economic results in power system steady state operations ever since. Basically, the equal incremental cost rule states that a lossless power system is optimally dispatched at a point where the incremental fuel cost of each generator,

\[ \frac{\partial f}{\partial P_j} \quad (j=1...N+1) \]

is equal to that of every other generator.

The system losses were later added to the problem formulation because the equal incremental cost rule suffers from calculation inaccuracy. This led to the development of what is called the B-coefficient method [1]. In this method, it is required that a relationship between the system losses of a power system and the real
power generation at each generation bus be established in order to calculate the equal incremental constant (see appendix A). In the late 1950's and the 1960's, work was undertaken to improve the loss formula representation. This occurred at the same time that the digital computer was first used to calculate power flow.

The B-coefficient method, although widely used after its appearance, possesses inherent drawbacks, mainly because the Kirchhoff Current and Voltage Laws are not included in the formulation of the method. Another problem in using the B-coefficient method is that it limits the introduction of the constraints imposed by power system operations and equipment, such as voltage, reactive power output constraints of generators and other constraints.

Mathematical achievements on optimization in the 1950's and the 1960's [3] stimulated the development of the Optimal Power Flow. In the early 1960's, much work was done to improve the problem formulation based on the Kuhn-Tucher theorem of nonlinear programming [23]. Among the pioneers, Carpentier's formulation plays a significant role in the progress in this field [4], [23]. The power flow solution technique used by Carpentier was Gauss-Seidel's method. The convergence behavior proved to be very slow. Around the same time, the Newton-Raphson method was being investigated to solve the power flow problems and shown to have very fast convergence properties [5]. The method became widely accepted after Tinney and others developed a very efficient, sparsity-programmed, ordered elimination method [6]. In the late 1960's, Dommel and Tinney extended
their Newton-Raphson method to the Optimal Power Flow [7]. It is well recognized that reference [7] lays a solid foundation in the development of the Optimal Power Flow problem. From then on, numerous papers concerned with this field have been published, aiming mainly at improving optimization algorithms for fast computation [10]-[20], decomposing the Optimal Power Flow problem formulation for computation efficiency and on-line applications [14], [16], [24], [25], adding additional operational constraints from system security considerations [18]-[21], [24], etc. Each of these aspects is reviewed below.

In the late 1960's, 1970's and early 1980's, various kinds of nonlinear programming methods, such as the Quasi-Newton, the DFP (the Davidon-Fletcher-Powell), the Conjugate Gradient, etc. were used in the place of the Steepest Descent method as used by Dommel and Tinney [7]. The problem of whether or not the penalty function method should be utilized in solving the constrained optimization problem was also investigated. As a result, methods that do not use the penalty function in handling the Optimal Power Flow problem involving inequality constraints became available [17], [18]. Much work was also done in the way in which the exact OPF problem (nonlinear programming problem) was approximated by its linear or quadratic forms in order that certain mathematical programming techniques, such as quadratic programming, could be adapted to solve them. These methods are claimed to have better convergence properties, less computer storage requirements and less computation time.
The decoupling characteristic between real power, bus voltage angle (or P-θ), and reactive power, bus voltage (or Q-V), led to the appearance of the so-called Real and Reactive Power, or P-Q Decomposition Optimal Power Flow [9], [14], [16]-[17], [24]. This approach enables the two OPF problems, the economic dispatch and the optimal reactive power allocation, to be completely decoupled from each other. With this method, the complete OPF problem can be solved in two stages. First, the economic dispatch problem is solved, holding the generation bus voltage unchanged. When the optimum (a location of real power generation at all generation buses) is found, the second stage of optimization starts. During the second stage optimization process, the total system transmission losses are minimized with respect to the voltage magnitudes at all generation buses, while keeping the real power generation constant. The feature of this method is that during each stage, only a small number of variables, rather than all the variables, are handled. As a result, mathematical programming techniques can be more efficiently used in finding the minimum. The method is well suited for on-line applications [9], [16], [17], [24].

With the development of the Optimal Power Flow, security constraints of power systems were added in the early 1970's [18]-[21]. This had a profound influence on the Optimal Power Flow as a whole. The Optimal Power Flow with security constraints requires that the solution to the problem be feasible not only for the normal operation of the system, but also for the situation under contingency conditions as well. The incorporation of security constraints into the
Optimal Power Flow usually represents much more work, because the problem now involves a large number of variables.

The use of modern mathematical programming techniques, especially the nonlinear programming approach, to solve power system problems has been developed, and various kinds of improvements have been made in the last 15 years. This paper presents a substantial review of these achievements. To limit the scope of this paper, only static power system optimization problems are considered. Furthermore, of these problems, the economic dispatch problem is mainly studied. The search of literature is also restricted to IEEE papers in IEEE Transactions on Power System Apparatus and Systems, PICA Proceedings, IEEE Proceedings and IEE Proceedings.

The search of the above journals revealed that numerous important publications had appeared on this topic since 1968. Almost every one of them has a slightly different OPF problem formulation from the others. It would be hard work to present all kinds of problem formulations and analyze them in detail. Instead, this paper presents only the most general OPF problem formulation under which the problem is solved. Other formulation methods are mentioned when necessary. Although there exist countless papers on this topic, from the mathematical programming point of view, the methods that are utilized to solve the Optimal Power Flow problem are only a few. For this reason, this paper summarizes the Optimal Power Flow solution methods based not on each individual paper but on the mathematical point of view.
The Optimal Power Flow problem is a constrained minimization problem involving both the equality and the inequality constraints. In order to make the solution procedure clear, the OPF problem is treated in two stages in this thesis. First, the inequality constraints are disregarded. The OPF problem is reduced to the one involving the equality constraints only. After all the basic approaches of solving this type of the problem have been presented, the inequality constraints are added to the original problem and methods of solving the complete OPF problem are given. These methods are the extensions of the basic approaches shown in the first stage.
SECTION 2. THE NATURE OF THE OPTIMAL POWER FLOW PROBLEM

The Optimal Power Flow problem was established based upon the load flow problem. A power system is basically composed of generation plants, transmission lines and loads. During steady state operation, the behavior of the power system is described by the nodal equations of the system. At each node or bus there are four quantities used to represent system operation, namely voltage V, phase angle $\theta$, real power $P$ and reactive power $Q$. Of these four quantities, two are specified and the other two are found.

Traditionally, buses are divided into three groups. Group 1 is the reference bus at which $V$ and $\theta$ are specified and $P$ and $Q$ are calculated. Without exception, every power system has only one reference bus. Group 2 is composed of generation buses (or $P,V$ buses) at which $P$ and $V$ are given and $Q$ and $\theta$ must be found. Group 3 consists of load buses (or $P,Q$ buses) at which $P$ and $Q$ are given and $V$ and $\theta$ are computed. It is assumed that the power system under investigation contains $N$ buses, of which the $N$th bus is denoted as the reference bus, and there are $N_g P,V$ buses and $N_L P,Q$ buses. The power flow computation is carried out to solve the nodal equations according to the specified quantities at each kind of bus. Table 1. shows the four types of quantities at three different kinds of buses. Because of the nonlinearity of the power flow equations, it is necessary to solve them iteratively. The widely used methods are the Newton-Raphson [5] and its modified version, the $P-Q$ decom-
position method [8], [9].

Table I. Four kinds of quantities at three types of buses

<table>
<thead>
<tr>
<th>Quantities</th>
<th>Reference bus</th>
<th>Generation buses</th>
<th>Load buses</th>
</tr>
</thead>
<tbody>
<tr>
<td>Knowns</td>
<td>$V_N, \theta_N$</td>
<td>$P_j, V_j$</td>
<td>$P_i, Q_i$</td>
</tr>
<tr>
<td>Unknowns</td>
<td>$P_N, Q_N$</td>
<td>$Q_j, \theta_j$</td>
<td>$V_1, \theta_1$</td>
</tr>
</tbody>
</table>

It should be noted that, given one set of known quantities in Table I., the unknown quantities can be uniquely determined by the power flow equations. Since the known quantities can be assumed to be any number, it turns out that there are an infinite number of solutions to the power flow equations which form a solution set $R$. In the solution set $R$ a subset $T$ ($T \in R$) can be defined which is physically meaningful. In the later sections of this paper, the power flow solutions are considered within the subset $T$.

The Optimal Power Flow solution is one of the power flow solutions from the subset $T$ which realizes an objective. This objective is achieved by optimally choosing one set of known quantities, also known as control variables, in the power flow equations and calculating the unknown quantities, known as state variables. The objective can be the one that gives less power production cost or less power transmission loss or both. It can also be something else. Minimization of the power generation cost is often referred to as the economic dispatch problem or P-$\Theta$ problem. Minimization of the system transmission losses is often referred to as the optimal reactive allocation problem or Q-V problem. They are subproblems of the
Optimal Power Flow problem. In power system operation under steady state conditions, the real power flows are strongly associated with the bus voltage angles $\theta$; and, likewise, the reactive power flows are associated with the bus voltage magnitudes $V$. The coupling between $P-\theta$ and $Q-V$ variable sets is comparatively weak. Therefore, the two subproblems can be treated either separately or simultaneously. In this paper, the economic dispatch problem or the $P-\theta$ problem is mainly studied.

As stated above, the objective here is to minimize the fuel cost of power generation while the power flow and other constraints imposed by the generation units and the transmission network of the power system are taken into account. When the total fuel cost of the power system is minimized, the real power generation among all the in-service generators is optimally allocated. In addition, the voltage levels and the reactive power outputs of the generators and transformer tap settings can also be determined. All these results are extracted using modern mathematical programming approaches. It is seen that in using the mathematical programming approach to solve the economic dispatch problem, some additional benefits can also be obtained. By contrast, traditional real power scheduling, although simple and still widely used, possesses incompleteness in the problem formulation and solution.

The following simplifications are made in the classic economic dispatch problem:

(1) Real power balance of the whole system is considered
totally, not individually for each bus, that is:

\[ P_L - P_G + P_{TL} = 0 \]

where: \( P_L \) stands for the total real power demand by the load
\( P_G \) represents the total power generated by the generators in the power system.
\( P_{TL} \) is the total system transmission losses.

(2) Reactive power injections to buses are disregarded.

(3) Only real power generations are variables and active during the computation process.

(4) Operational limits, except the real power generation, of both the power system and the equipment are ignored.

Under these assumptions, the solution to the economic dispatch problem is the optimal allocation of real power among all the generators in the system subject to the upper or lower limit of real power output of each generator.

It should be pointed out that, because the modern Optimal Power Flow problem is formulated and solved on a firm mathematical basis, one may ask for additional savings through the rigorous study of the problem. A comparison study was undertaken by Happ [23] aimed at determining the financial benefits of changing from the classic economic dispatch to a more advanced method of dispatching. The test system used was an 118 bus IEEE system whose generator characteristics were considered to be quite typical. The study shows that from an economical point of view, the classic technique does as good a job as the rigorous method; so long as the B matrix of the B-
coefficient method (appendix A) is updated along with the means of the system operation. The reason for employing more advanced techniques is, then, that more rigorous models are required for executing different functions associated with the security of power system operations; not for getting additional savings. The power system security operation calculations can be performed by introducing various kinds of equality and inequality constraints in the Optimal Power Flow solutions. This is the other important application of the Optimal Power Flow studies.
SECTION 3. FORMULATION OF THE OPTIMAL POWER FLOW PROBLEM

According to the definition of the Optimal Power Flow problem, its formulation is as follows:

Minimize the total cost of power generation in the power system subject to the inequality constraints of:

1. Voltage magnitude $V_j$ at the jth P,V bus ($j=1...N_g$).
2. Real power $P_j$ at the jth P,V bus ($j=1...N_g$).
3. Tap $T_k$ of each tap-changing transformer ($k=1...N_t$).
4. Reactive power $Q_i$ at the ith P,Q bus ($i=1...N_L$).
5. Voltage magnitude $V_i$ at the ith P,Q bus ($i=1...N_L$).

subject to the equality constraints of the power flow equations.

(a)

It is assumed that the operating cost function of every generator $K(P_j)$ is in a quadratic form:

At the P,V buses,

$$K(P_j) = (A_j + B_j P_j + C_j P_j^2 ), \quad j=1...N_g$$  \hspace{1cm} (3.1)

At the reference bus,

$$K(P_N) = (A_N + B_N P_N + C_N P_N^2 )$$  \hspace{1cm} (3.2)

where: $A_j$, $B_j$, $C_j$ are constants associated with the cost function of the jth generator, ($j=1...N_g$);
$A_N$, $B_N$, $C_N$ are constants associated with the cost function of the generator of the reference bus;
$P_j$ is the real power output at the jth P,V bus ($j=1...N_g$);
\( P_N \) stands for the real power generation at the reference bus; 
\( N_g \) represents the number of P,V buses of the power system;

The total cost of power generation of the power system is the sum of the power generation costs of all in-service generators.

\[
 f = \left[ \sum_{j=1}^{N} g_j(P_j) \right] + K(P_N) = \left[ \sum_{j=1}^{N} (A_j + B_j P_j + C_j(P_j)^2) \right] \\
+ (A_N + B_N P_N + C_N P_N^2) 
\]  

(3.3)

\( f \) is the objective function which is going to be minimized to achieve the best economic benefit.

The optimal flow problem can then be mathematically restated as:

\[
\text{Min } f = \text{Min } \left[ \sum_{j=1}^{N} g_j(P_j) \right] + K(P_N) \\
= \text{Min } \left[ \sum_{j=1}^{N} (A_j + B_j P_j + C_j(P_j)^2) \right] + (A_N + B_N P_N + C_N P_N^2) 
\]  

(3.4)

with respect to \( P_j \) and \( P_N \)

The minimization is subject to the inequality constraints of, viz;

\begin{align*}
(1) & \quad \min \max \quad V_j \leq V_j \leq V_j \quad \text{at each P,V bus (} j=1 \ldots N_g \text{).} \\
(2) & \quad \min \max \quad P_j \leq P_j \leq P_j \quad \text{at each P,V bus (} j=1 \ldots N_g \text{).} \\
(3) & \quad \min \max \quad Q_j \leq Q_j \leq Q_j \quad \text{at each P,V bus (} j=1 \ldots N_g \text{).} \\
(4) & \quad \min \max \quad V_i \leq V_i \leq V_i \quad \text{at each P,Q bus (} i=1 \ldots N_L \text{).} \\
(5) & \quad \min \max \quad T_k \leq T_k \leq T_k \quad \text{for each tap-changing transformer (} k=1 \ldots t \text{).}
\end{align*}
The minimization is also subject to the equality constraints of the power flow equations, viz;

\[
P_{Gm} = P_m = P_{Lm} + P_{INm} \quad m=1\ldots N-1 \quad (3.5)
\]

\[
Q_{Gn} = Q_n = Q_{Ln} + Q_{INn} \quad n=1\ldots N-1-N_g \quad (3.6)
\]

In (3.5), \(P_{Gm}=P_m\) is the real power generated at the \(m\)th P,V bus, \(m=1\ldots N\). \(P_{Gm}=0\), for \(N_g<m<(N-1)\). \(P_{Lm}\) and \(P_{INm}\) represent the real power load at the \(m\)th bus and the real power injected into the \(m\)th bus respectively, \(m=1\ldots N-1\). In (3.6), \(Q_{Gn}=Q_n\) is the reactive power generated at the \(n\)th P,V bus, \(n=1\ldots N_g\). \(Q_{Ln}\) and \(Q_{INn}\) stand for the reactive power load at the \(n\)th bus and the reactive power injected into the \(n\)th bus, respectively, \(n=1\ldots N-1-N_g\).

In the Optimal Power Flow computations, it is convenient to define two sets of variables; namely, \([X]\) which is called the state variable vector and \([Z]\) which is the control variable vector.

The vector \([X]\) is defined as:

\[
[X] = \begin{bmatrix}
V_i \\
\theta_i \\
\theta_j
\end{bmatrix}
\text{on each P,Q bus}
\]

\[
\text{on each P,V bus}
\]

The real power \(P_N\) at the reference bus is also an unknown quantity which should appear in the \([X]\) vector; but it can be seen from the equation

\[
P_{GN} = P_N = P_{LN} + P_{INN} \quad (3.7)
\]

that \(P_N\) is a function of both the control and the state variables. \(P_N\) eventually does not appear in the \([X]\) vector.
The vector $[Z]$ is defined as:

$$[Z] = \begin{bmatrix}
    P_j \\
    V_j \\
    T_k \\
    V_N \\
    \theta_N \\
    P_i \\
    Q_i
\end{bmatrix}$$

- $P_j$, $V_j$ on each $P,V$ bus
- $T_k$ for each tap-changing transformer
- $V_N$ on the reference bus
- $\theta_N$
- $P_i$ on each $P,Q$ bus
- $Q_i$

Note that in the $[Z]$ vector, $V_N$, $\theta_N$, $P_i$ and $Q_i$, $(i=1\ldots N_L)$ are all fixed parameters, therefore, the $[Z]$ vector can be further partitioned into two subvectors $[U]$ and $[P]$ such that:

$$[Z] = \begin{bmatrix}
    [U] \\
    [P]
\end{bmatrix}$$

where: $[U]$ is the control subvector

$$[U] = \begin{bmatrix}
    P_j \\
    V_j \\
    T_k
\end{bmatrix}$$

at each $P,V$ bus

for each tap-changing transformer

$[P]$ is the fixed parameter subvector

$$[P] = \begin{bmatrix}
    V_N \\
    \theta_N \\
    P_i \\
    Q_i
\end{bmatrix}$$

at the reference bus

at each $P,Q$ bus

The dimensions of the vectors $[X]$, $[U]$ and $[P]$ are:
[X] is of the dimension \((N_g + 2N_L)\)

[U] is of the dimension \((2N_g + N_t)\)

[P] is of the dimension \(2(N_l + 1)\)

In the later sections of this paper, minimization problems are solved with respect to the control subvector \([U]\). After the state and the control vectors \([X]\) and \([U]\) have been defined, the equality constraints (3.5) (3.6) (the power flow equations) can be rewritten explicitly as:

\[
[G(X, U)] = [0]
\]  \hspace{1cm} (3.8)

where: \([G(X, U)]\) represents a vector function of dimension \((2(N_l) - N_g)\) by 1.

The inequality constraint sets on page 13-14 can be split into two parts. The first part is made of control variable constraints which are voltage magnitudes \(V\), real and reactive powers \(P, Q\) at all \(P, V\) buses, and taps \(T\) of tap-changing transformers. The second part is composed of state variable constraints, called functional constraints. There are voltage magnitudes \(V\) at all \(P, Q\) buses. Both sets of constraints can be expressed concisely by the following notation:

\[
[H(X, U)] \leq [0]
\]  \hspace{1cm} (3.9)

The above expression represents a total number of inequality constraints equal to \(4(N_l + N_g) + 2N_t\).

# From here on, for simplicity the control subvector \([U]\) will also be called the control vector.
It should be noted that there are other inequality constraints, such as real and reactive power interchanges between areas, real and reactive power flows in transmission lines etc., that can also be added to \( [H(X,U)] \leq 0 \).

With the definitions above, the optimal power flow problem (a) can now be written in a compact form:

\[
\begin{align*}
\text{Min} & \quad f(X, U) \\
\text{s.t.} & \quad [G(X, U)] = [0] \\
& \quad [H(X, U)] \leq [0]
\end{align*}
\]

This is a standard nonlinear programming problem with both equality and inequality constraints. In the next sections, solution techniques are presented.
SECTION 4. THE OPTIMAL POWER FLOW SOLUTIONS

4.1 THE OPTIMAL POWER FLOW SOLUTIONS USING NONLINEAR PROGRAMMING METHODS

So far, Optimal Power Flow problems have been solved by various kinds of nonlinear programming methods. Mathematically, the nonlinear programming methods belong to one of the two categories. One category of methods utilizes the derivative information of the objective function, (known as the gradient) to search for descent direction in order to decrease the value of the objective function. The other category of methods, which has not been used widely so far for Optimal Power Flow problems, calculates the values of the objective function directly without computing the derivatives, followed by comparison of each of the proposed values of the objective function to find a better value of the variables of the function. Because of the complexity of the objective function for this problem, both categories of methods need many iterations, and computer time consumption is very high. In the first category of methods, the second derivatives of the objective function are sometimes required by certain kinds of optimization methods. In the next subsections, nonlinear programming methods which use the derivative information of the objective function to solve the OPF problems are reviewed.
4.1.1 The Optimal Power Flow Solution Without

The Inequality Constraints

The Optimal Power Flow problem without the inequality constraints is:

Min \( f(X, U) \)

s.t. \( [G(X, U)] = [0] \) \hspace{1cm} (c)

The problem can be solved by means of forming an augmented objective function \( L(X, U, \lambda) \) by introducing a Lagrangian multiplier vector \( [\lambda] \) to make the equality constrained problem an unconstrained one [2], (appendix c). We make the following definition:

\[
L(X, U, \lambda) = f(X, U) + [\lambda]^t [G(X, U)]; \tag{4.1}
\]

where: \( [\lambda]^t \) is the transposition of the vector \( [\lambda] \) which has the same dimension as that of \( [G(X, U)] \).

The minimum point occurs where:

\[
\left[ \begin{array}{c}
\frac{\partial L}{\partial [X]} \\
\frac{\partial L}{\partial [U]} \\
\frac{\partial L}{\partial [\lambda]}
\end{array} \right] = \left[ \begin{array}{c}
\frac{\partial f}{\partial [X]} \\
\frac{\partial f}{\partial [U]} \\
\frac{\partial [G]}{\partial [\lambda]}
\end{array} \right] + \left[ \begin{array}{c}
\frac{\partial [G]}{\partial [X]} \\
\frac{\partial [G]}{\partial [U]} \\
\frac{\partial [G]}{\partial [\lambda]}
\end{array} \right]^t [\lambda] = [0] \tag{4.2}
\]

\[
\left[ \begin{array}{c}
\frac{\partial L}{\partial [X]} \\
\frac{\partial L}{\partial [U]} \\
\frac{\partial L}{\partial [\lambda]}
\end{array} \right] = \left[ \begin{array}{c}
\frac{\partial f}{\partial [X]} \\
\frac{\partial f}{\partial [U]} \\
\frac{\partial [G]}{\partial [\lambda]}
\end{array} \right] + \left[ \begin{array}{c}
\frac{\partial [G]}{\partial [X]} \\
\frac{\partial [G]}{\partial [U]} \\
\frac{\partial [G]}{\partial [\lambda]}
\end{array} \right]^t [\lambda] = [0] \tag{4.3}
\]

\[
\left[ \begin{array}{c}
\frac{\partial L}{\partial [\lambda]} \\
\frac{\partial L}{\partial [X]} \\
\frac{\partial L}{\partial [U]}
\end{array} \right] = [G(X, U)] = [0] \tag{4.4}
\]
In the above equations, the derivatives of the scalar function \( L(X,U,\lambda) \) are taken with respect to three vectors \([X]\), \([U]\) and \([\lambda]\) (appendix B), resulting in three sets of generally nonlinear equations, which provide all the stationary points in the region of interest. Due to the convex property of the objective function \( (f(X,U) \) is nearly a quadratic function), it is easy to show that once the stationary points \( [X^s], [U^s], [\lambda^s] \) are obtained, they are the global minimum of \( f(X,U) \). It can be recognized that

\[
\begin{bmatrix}
\frac{\partial f}{\partial [U]}
\end{bmatrix}
\]

is the gradient of \( f(X,U) \) while holding the equality constraints, and

\[
\begin{bmatrix}
\frac{\partial [G]}{\partial [X]}
\end{bmatrix} = [J]
\]

is the Jacobian matrix from the power flow studies in the use of the Newton-Raphson method.

Theoretically, to get the optimum points \( [U]=[U^*], [X]=[X^*], [\lambda]=[\lambda^*] \), it is required to solve (4.2), (4.3) and (4.4) simultaneously. However, due to its high dimensionality and the nonlinearity of the problem, the above equations are usually solved iteratively. The iteration process starts with assuming a control vector \([U]^0\) and solving for \([X]^0\) through the study of the power flow equations (4.4). The purpose of the above step is to provide a feasible starting point \([X]=[X]^0, [U]=[U]^0\) (but not yet an optimal point) for the solution of equations (4.2) and (4.3). When the initial values \([X]^0\) and \([U]^0\) are available, the Lagrangian multiplier vector \([\lambda]^0\) is computed using (4.2).
Up to this point, equations (4.2) and (4.4) have been satisfied. In fact, (4.2) is forced to be satisfied by selecting

\[
[\lambda]^0 = - [(J^t)^{-1} \left[ \frac{\partial f}{\partial [X]} \right]]
\]  

(4.5)

The calculation of \( \left[ \frac{\partial L}{\partial [U]} \right] \) is then carried out by substituting (4.5) into (4.3), that is:

\[
\left[ \frac{\partial L}{\partial [U]} \right] = \left[ \frac{\partial f}{\partial [U]} \right] - \left[ \frac{\partial [G]}{\partial [U]} \right]^t \left[ \left( \frac{\partial [G]}{\partial [X]} \right)^t \right]^{-1} \left[ \frac{\partial f}{\partial [X]} \right]
\]  

(4.6)

The procedure for checking whether \( \left[ \frac{\partial L}{\partial [U]} \right] \) is zero or less than a previously given tolerance \( \varepsilon \) is then performed. If the requirement is met at this point, the optimal point has been reached. If not, improvement of the vector \([U]\) is made and the above procedure is repeated. Let \( k \) represent the \( k \)th iteration. The algorithm is summarized below:

(a) Assume a set of initial control parameters \([U]^0\), usually the real powers \( P \) and the voltage magnitudes \( V \) at all \( P,V \) buses. The real and the reactive powers \( P \) and \( Q \) at all \( P,Q \) buses and \( V_N, \theta_N \) at the reference bus are fixed during the compu-
(b) Calculate the controlled, or the state, vector 
\[ X^k \]
from the power flow equations using \( U = U^k \), \( k=0,1,...,n \). The results are the voltage magnitudes at all \( P,Q \) buses and the voltage angles \( \theta \) at all the \( P,Q \) and \( P,V \) buses. Store the Jacobian matrix \( J \) at \( [X] = [X]^k \), \( [U] = [U]^k \), \( k=0,1,2...n \).

(c) Compute the Lagrangian multiplier vector \( \lambda^k \) from (4.2) at \( [X] = [X]^k \), \( [U] = [U]^k \), \( k=0,1,...,n \).

(d) Compute the gradient of \( L(X,U,\lambda) \) with respect to vector \( [U] \), that is:

\[
\left[ \frac{\partial L}{\partial [U]} \right] = \left[ \frac{\partial f}{\partial [U]} \right] - \left[ \frac{\partial [G]}{\partial [U]} \right]^t \left[ \frac{\partial [G]}{\partial [X]} \right]^{-1} \left[ \frac{\partial f}{\partial [X]} \right]
\]

at \( [X] = [X]^k \), \( [U] = [U]^k \) and \( [\lambda] = [\lambda]^k \) for \( k=0,1,...,n \).

(e) Check if \( \left[ \frac{\partial L}{\partial [U]} \right] \leq \epsilon \). If the above relation is satisfied, \( [U]^k \) is the optimal point; otherwise set \( k=k+1 \) and go back to step (f).

(f) Update \( [U]^{k+1} = [U]^k + [\delta U]^k \) then go back to step (b).

A flow chart of the algorithm is given in Fig. 1 which shows all the steps above (see next page).
Input data

Assume initial control vector \([U]=[U]_0\)

Solve for power flow equations \([G(X,U)]=[0]\) to get \([X]_k^{}, k=0,1...n\)

Compute \([\lambda]_k^{}\) at \([X]=[X]_k^{}, [U]=[U]_k^{}\), \(k=0,1...n\)

Calculate gradient of \(L(X,U,\lambda)\)

\(k=k+1\)

Gradient of \(L<\epsilon\)?

Yes

Compute \([\delta U]\)

Optimum reached stop

No

compute \([U]^{k+1}= [U]_k^{} + [\delta U]_k^{}\)

Iteration 1

Iteration 2

Fig 1. A simplified flow chart of the algorithm
It is seen from the above steps that the algorithm is centered around solving the power flow equations to provide a sequence of feasible points. By this it is meant that the equality constraints must always be satisfied during the process of searching for the optimal control vector $[U]$.

It is clear that the initial values of $[X]$ and $[U]$ act only as a means of starting the algorithm. Iteration is usually required until the optimum is reached. Since the Newton-Raphson method is the most powerful technique for solving the power flow equations, the computational efficiency of the above algorithm depends largely upon the improvement of $[\delta U]^k$ in step (f). There are two aspects involved in improving $[\delta U]^k$. One is the direction of change of the objective function and the other is how large the step length is going to be to greatly reduce the values of the objective function. Small progress of $[\delta U]$ leads to slow convergence; while big changes sometimes result in failure of convergence.

Mathematically, there are several methods of updating $[\delta U]$. Among them, those that are based on the first order or second order derivative information are widely used.

(1) The Steepest Descent Method

The Steepest Descent method was the first method used in the Optimal Power Flow computations. The updated control parameter $[U]$ is calculated as shown below:
Suppose at the kth iteration \((k=0,1\ldots n)\), the control vector \([U]\) is \([U]^k\). During the next iteration \((k=k+1)\), \([U]\) is updated by:

\[ [U]^{k+1} = [U]^k + [\delta U]^k \]  

(4.7)

According to the Steepest Descent method, \([\delta U]^k\) is given in the following form:

\[ [\delta U]^k = -B^k \left[ \frac{\partial f}{\partial [U]} \right]^k = -B^k [g]^k \]  

(4.8)

In the above expression, \([g]^k\) is recognized as the gradient of the function \(f(X,U)\) at \([X]=[X]^k\), \([U]=[U]^k\) while holding \([G(X,U)]=[0]\). The negative value of the gradient points out the descent direction of \(f(X,U)\). \(B^k\) is a scalar parameter to be determined in order to greatly reduce the function numerically along the negative direction of \([g]^k\).

The procedure for selecting \(B^k\) in the given direction \([g]^k\) is as follows:

Let the current iteration be \(k\) and \([U]=[U]^k\), \([X]=[X]^k\) and \([\lambda]=[\lambda]^k\). Before the next iteration \((k=k+1)\) starts, the scalar \(B^k\) is computed by forming the following function:

\[ L(\lambda^k,(U^k - B^k g^k),\lambda^k) = f(X^k,(U^k - B^k g^k)) + [\lambda^k]^t [G(x^k,u^k)] \]  

(4.9)

Notice that (4.9) is a scalar function of \(B^k\). The value of \(B^k\) can be found such that the function \(L\) has the minimum value along the negative direction of \([g]^k\).

\[ \frac{dL}{dB^k} = \frac{df}{dB^k} + \left[ \frac{d[G]}{dB^k} \right]^t [\lambda]^k = 0 \]  

(4.10)
After $B^k = B_{\text{min}}^k$ has been found, adjustment of $[U]$ is set to:

$$[U]^{k+1} = [U]^k - B_{\text{min}}^k [g]^k$$  \hspace{1cm} (4.11)

This procedure is often referred to as minimization in a given direction. Numerically, $B^k$ is found using the "Golden Ratio" method [2], [3].

$[\delta U]^k$ can also be computed directly by a means of taking advantage of the second derivative information of the cost function $f(X,U)$ [7]:

$$[\delta U] = - [A^k]^{-1} [g]^k$$  \hspace{1cm} (4.12)

where: $[A] = \left[ \frac{\partial^2 f}{\partial u_i \partial u_j} \right]$ is the Hessian matrix of second derivatives.

$[U]^{k+1}$ is then given by

$$[U]^{k+1} = [U]^k + [\delta U]^k = [U]^k - [A^k]^{-1} [g]^k$$  \hspace{1cm} (4.13)

If $f(X,U)$ were a truly quadratic function, $[U]^k$ would be the optimal point, otherwise, iterations are necessary. It was shown in [7] that finding the Hessian matrix and solving the above equation to get the vector $[\delta U]^k$ results in considerably more computer time per cycle than first-order gradient methods. It can fail to converge if the Hessian matrix is not positive definite [2]. To overcome the difficulty, an approximated second order method is used in [7]. The method neglects all the off-diagonal elements in the Hessian matrix, provided that the control parameters have no (or little) interaction. The main diagonal elements of the Hessian matrix are
computed from a small exploratory displacement in vector \([U]\).

\[
\left[ \frac{\partial^2 f}{\partial u_j^2} \right] \approx \frac{\text{change in } \left[ \frac{\partial f}{\partial u_j} \right]}{\text{change in } u_j} = \left[ \frac{\partial f}{\partial u_j} \right]_{k-1} - \left[ \frac{\partial f}{\partial u_j} \right]_{k} (j=1\ldots2N_g+N_t) \tag{4.14}
\]

The rate of convergence characteristics of the Steepest Descent method, which uses mainly the first derivative information, proves to be linear \([2]\). Better convergence can be expected by using other methods.

(2) The Newton and the Quasi-Newton Methods

This method is used in \([7]\) only when the optimum is about to be reached, and also the second derivatives are approximated by their differences. In \([10], [11]\) the complete use of this method to adjust control vector \([U]\) can be found.

It is well known that at the minimum,

\[
\left[ \frac{\partial f}{\partial [U]} \right] = \text{grad } f(X^*, U^*, \lambda^*) = [g(X^*, U^*, \lambda^*)] = [0] \tag{4.15}
\]

where : \([X^*], [U^*], [\lambda^*]\) represent the minimum point.

Usually, the above equation represents a set of nonlinear equations. The following expression gives the iteration formula :

\[
[U]^{k+1} - [U]^k = [\delta U]^k = - \left[ \frac{\partial [g]^k}{\partial [U]^k} \right]^{-1} [g]^k
\]

\[
= \left\{ \left[ \frac{\partial^2 L}{\partial [U]^k \partial [U]^k} \right]^k \right\}^{-1} \left[ \frac{\partial f}{\partial [U]^k} \right]^k = - [A^k]^{-1} [g]^k \tag{4.16}
\]
or:

\[ [A]^k [\delta U]^k = - [g]^k \]  \hspace{1cm} (4.17)

In [10], instead of calculating the inverse Hessian at each iteration, \([\delta U]\) is found by Gaussian elimination using sparsity techniques (in the power system equations, the Hessian matrix of \([G(X,U)]\) is a sparse matrix, but not as sparse as the Jacobian matrix [10]). Although each iteration takes more time than that of the first gradient method, the whole process needs only a few iterations because the Newton's method is very powerful in solving nonlinear equations.

The success of this method is due to its use of Gaussian elimination using sparsity techniques to calculate the correction \([\delta U]\) instead of calculating the inverse of the Hessian matrix at each iteration, but the computation of the elements of the Hessian matrix is by no means easy. In [12]-[15], the modified Newton method (also known as the Quasi-Newton method) is used. The feature of the approach is that the Hessian matrix is never computed in updating control vector \([U]\). Instead, \([\delta U]\) is obtained by computing the difference matrix of the gradient of the objective function (\([g]^{k+1}_{k} - [g]^k\)) between two consecutive iterations and a matrix \([H]\), called the direction matrix, which approximates the Hessian matrix. The \([H]\) matrix is arbitrarily assumed at the beginning of the calculation and updated during the computation process. There exist several ways of constructing the \([H]\) matrix. Accordingly, there are several versions of the Quasi-Newton method. One of the most widely used methods in the Optimal Power Flow computations is the DFP method (the
The DFP Method

In this method, the iteration formulae are as follows:

$$[U]^{k+1} = [U]^k - B^k [H]^k [g]^k$$  \hspace{1cm} (4.18)

where:

$$[H]^{k+1} = [H]^k + [C]^k - [D]^k$$

$$[C]^k = \frac{[\delta U]^k ([\delta U]^k)^t}{([\delta U]^k)^t ([\delta g]^k)}$$

$$[D]^k = \frac{[H]^k [\delta g]^k ([\delta g]^k)^t ([H]^k)^t}{([\delta g]^k)^t ([H]^k)^t ([\delta g]^k)}$$

$$[\delta U]^k = [U]^{k+1} - [U]^k$$

$$[\delta g]^k = [g]^{k+1} - [g]^k$$

Prior to iteration, $[H]^0$ is simply chosen as an identity matrix $[I]$. Then at the first iteration, $k=0$, (4.18) can be written as:

$$[U]^1 = [U]^0 - B^0 [I]^0 [g]^0 = [U]^0 - B^0 [g]^0$$  \hspace{1cm} (4.19)

It is seen that the first step of the iteration is the Steepest Descent method. At the kth iteration, $[H]^{k+1}$ becomes:

$$[H]^{k+1} = [I] + \sum_{i=0}^{k} [C]^i - \sum_{i=0}^{k} [D]^i$$  \hspace{1cm} (4.20)

The above expression exhibits the fact that at the kth iteration, the direction matrix $[H]^{k+1}$ is computed utilizing all the previous information about $[C]^i$, $[D]^i$, $[\delta U]^i$ and $[\delta g]^i$, $i=0,1,...,k$.

Because of this, for the quadratic type of objective function,
two consecutive steps of iterations are conjugate [3]. As a result, the convergence property of the method is usually much better than that of the Steepest Decent method. The method was used to solve optimal power flow problems up to 100 variables [12], [13]. For the problems in which more than 100 variables are involved, the method may be time-consuming or even break down [10]. This is caused partially by the fact that the computation of the direction matrix \([H]\) gets more difficult with problem size and number of iterations. From the point of view of computation efficiency and robustness, some researchers preferred other methods that maintain the property of the DFP method, but require less computation.

(4) The Conjugate Gradient Method

This method is widely used in the Optimal Power Flow computations [15], [16] and [24].

The searching direction of this method at the \((k+1)th\) iteration is the linear combination of the negative gradient at the current point \([U] = [U]^{k+1}\) and the negative gradient at the previous point \([U] = [U]^k\)

\[
[P]^{k+1} = - [g]^{k+1} + \alpha^k [P]^k
\]

\[
\alpha^0 = 0
\]

where: \([P]^{k+1}\) is the searching direction at the \((k+1)th\) iteration, \([g]^{k+1}\) and \([g]^k\) are the gradient of \(L\) at the \((k+1)th\) and the \(kth\) iteration, respectively.

\([P]^k\) is the searching direction at the previous iteration.
\[ \alpha^k = \frac{\|g^{k+1}\|^2}{\|g^k\|^2} \quad \text{(if } L \text{ is quadratic)} \]

\[ \alpha^k = \frac{\|g^{k+1}\|^2 - (g^{k+1})^t g^k}{\|g^k\|^2} \quad \text{(if } L \text{ is nonlinear)} \]

\[ \|g\| \] represents the norm of the vector \([g]\).

At each iteration, after the searching direction has been obtained, minimization in the direction to find the optimal step \(B\) is usually performed.

For the \(n\)-dimensional, quadratic type of objective functions using this method, the optimal point can be obtained in \(n\) linearly independent steps [2]. For the \(n\)-dimensional, nonquadratic type of objective functions, the steps required to reach the optimum are more than \(n\).

A comparison among the three basic gradient methods is given through an example. In [15], the performance of the Steepest Descent, the DFP and the Conjugate Gradient methods is evaluated on an IEEE 118 bus system. It turns out that the DFP and the Conjugate Gradient performed well until the solution was approached. The rate of convergence using the Steepest descent algorithm was very slow.
4.1.2 The Optimal Power Flow Solutions Involving

The Inequality Constraints

In the last section, the solution of the optimal power flow problem (b)

$$\begin{align*}
\text{Min} & \quad f(X,U) \\
\text{s.t.} & \quad [G(X,U)] = [0] \\
& \quad [H(X,U)] \leq [0]
\end{align*}$$

was described in the case of no inequality constraints. Generally speaking, the solution is not acceptable, because it does not guarantee that all the control and state variables are within their limits. Therefore, it is important that the inequality constraints be included in the solution.

There are techniques available for the solution of the constrained nonlinear programming problem (b) [2]. All these methods can be classified into two broad categories; namely the direct method and the indirect methods. Of the indirect methods, the Gradient Projection method [2] (or the Rosen method) has been adopted recently to solve the problem given by (b) [17]. The direct method, or the Penalty Function method, is composed of the Interior Penalty Function method and the Exterior Penalty Function method [3]. Some of the related topics are presented below.
4.1.2.1 Solution Of (b) By The Penalty Function Method

The Optimal Power Flow problem was solved primarily by the Penalty Function method [22]. It is a basic approach in handling the constrained optimization problems.

The penalty function methods transform the original optimization problem (b) into alternative formulations such that numerical solutions are sought by solving a sequence of unconstrained minimization problems. This technique is often referred to as the Sequence of Unconstrained Minimization Technique, or SUMT. A sequence of unconstrained minimization problems can be obtained by adding a so-called penalty function into the basic objective function to form an augmented objective function. The sequential unconstrained minimization problems can then be solved with respect to the augmented objective function until the minimum point is found.

Basically, the penalty function method can handle both the equality and the inequality constraints simultaneously. Accordingly, there are two methods that can be used to form the augmented objective function. One way is to utilize Lagrange multipliers to treat the equality constraints \[ G(X,U) = [0] \] in (b) and to use the penalty function to deal with the inequality constraints \[ H(X,U) \leq [0] \] as used by many researchers [7], [11], [14]-[16] etc. The other method is to use the penalty function to handle both equality and inequality constraints as seen in references [10], [12], [13]. This method is
sometimes referred to as the pure penalty function approach.

The first method minimizes the following kind of augmented objective function:

\[
L(X,U,\lambda,c^k) = f(X,U) + [\lambda]^t [G(X,U)] + [W(X,U)]
\]

or equivalently,

\[
L(X,U,\lambda,c^k) = f(X,U) + \sum_{i=1}^{p} \lambda_{i} g_{i} + c^k \sum_{j=1}^{q} w_{j}
\]

where: \( p=2(N-1)-N_g \)

In (4.23), the first two terms on the right hand side of the above equation are the same as before. The term \( c^k \sum_{j=1}^{q} w_{j} \) is the sum of the squares of the \( q \) inequality constraints which have been violated in problem (b). \( c^k \) is the penalty parameter at the \( k \)th iteration.

\( w_j \) is defined as:

\[
w_{j} = \begin{cases} 
\min & (x_{j} - x_{j})^2 \quad \text{if } x_{j} \leq x_{j} \\
0 & \text{if } x_{j} \geq x_{j}
\end{cases}
\]

where: \( w_{j} \) is the \( j \)th penalty function associated with the \( j \)th component of the vector \([X]\) that has violated its upper or lower limit. \( w_{j} \) takes only one of its three forms, depending on the region in which the value of \( x_{j} \) fell during the last
iteration.

In the second method, the augmented objective function is constructed in the following form:

$$L(X,U,b^k,c^k) = f(X,U) + b^k \sum_{i=1}^{m} g_i^2(X,U) + c^k \sum_{j=1}^{n} w_j$$

(4.25)

where: $$[g_i(X,U)]^2$$ is the square of the $i$th equation from the power flow equation set.

$b^k$ and $c^k$ are the penalty parameter constants ($k=0,1..n$).

$f(X,U)$ is the same as before.

The way in which the penalty function $[W(X,U)]$ is constructed is based on Zangwill's "Exterior SUMT" method (see appendix D).

In both cases, a term $[W(X,U)]$ is added to the augmented objective function to penalize the violation of constraints by means of adding additional value to the objective function. When the control parameter $[U]$ is such that both the second and the third term vanish, the value of $L(X,U,\lambda,c^k)$ or $L(X,U,b^k,c^k)$ is equal to that of $f(X,U)$.

The solution procedure with the penalty function involved is basically the same as the case where the inequality constraints are not considered. Let $L(X,U,\lambda,c^k)$ be:

$$L(X,U,\lambda,c^k) = f(X,U) + \sum_{i=1}^{p} \lambda_i g_i(X,U) + c^k \sum_{j=1}^{q} w_j$$

(4.26)

To begin with iteration, $c^k$ must be given an initial value, say $c^0=1$. Notice that after $c^k$ has been numerically given, the above problem becomes an unconstrained nonlinear minimization problem.
Theoretically speaking, any nonlinear programming method previously given can be employed to solve the problem.

The necessary conditions for achieving a minimum are:

\[
\begin{align*}
\left[ \frac{\partial L}{\partial [\lambda]} \right] &= [G(X,U)] = [0] \\
\left[ \frac{\partial L}{\partial [X]} \right] &= \left[ \frac{\partial f}{\partial [X]} \right] + \left[ \frac{\partial [G]}{\partial [X]} \right]^t [\lambda] + \left[ \frac{\partial [W]}{\partial [X]} \right] = [0] \\
\left[ \frac{\partial L}{\partial [U]} \right] &= \left[ \frac{\partial f}{\partial [U]} \right] + \left[ \frac{\partial [G]}{\partial [U]} \right]^t [\lambda] + \left[ \frac{\partial [W]}{\partial [U]} \right] = [0]
\end{align*}
\]

In the above equations, the second set of equations can always be automatically satisfied, provided that the vector \([r]\) is chosen such that:

\[
[\lambda] = - \left[ \left( \frac{\partial [G]}{\partial [X]} \right)^t \right]^{-1} \left\{ \left[ \frac{\partial f}{\partial [X]} + \frac{\partial [W]}{\partial [X]} \right] \right\}
\]

When vector \([\lambda]\) has been found, the gradient of \(L\) is calculated. Normally, \[\frac{\partial L}{\partial [U]}\] is not zero at this point (the optimum has not been reached). Adjustments of the control vector \([U]\) must be implemented. The improved control vector is assumed to be:

\[\[U\]^{k+1} = [U]^k + [\delta U]^k\]

The next step is to sort out those components of the vector \([U]^{k+1}\) which exceed the limits of either \([U]^{\text{min}}\) or \([U]^{\text{max}}\). During the next iteration, the values of those components are set to the corresponding limits. The corresponding state vector \([X]^{k+1}\) is obtained through performing another study of the power flow equations.
Like the treatment of the control vector, each component of the state vector \([X]^{k+1}\) is then checked to find out those that have exceeded their limits. The penalty function associated with each of these components of the state vector \([X]\) is then formed with an increased value of \(c^{k+1} = a c^k\), \((a>1)\). During the next iteration, the control vector \([U]^k\), the state vector \([X]^k\) are replaced by \([U]^{k+1}\) and \([X]^{k+1}\) respectively, and the augmented objective function is also substituted by the newly formed one. The above process is repeated until the gradient of \(L\) is less than the previously given tolerance \(\epsilon\) while all the equality and the inequality constraints are simultaneously satisfied.

A comparison between the optimization process without the inequality constraints and that with the inequality constraints reveals two important differences. One is that, in the latter, the procedure of checking whether or not each component of the control and the state vector has exceeded its limit has to be carried out. If any component of the state vector \([X]\) is beyond its limit, it's corresponding penalty function is not zero. This results in some additional values to the original objective function \(f(X,U)\). This means that the optimum has not been reached. Another difference is the addition of selecting the penalty weighting factor \(c^k\) varying from its initial \(c^0\) value to an infinitely large one. During this process, a sequence of unconstrained minimization problems are solved until the minimum point which satisfies all the equality and inequality constraints is reached.
The Optimal Power Flow problem has also been solved by the pure penalty function method [10], [12], [13]. The formulation of the Optimal Power Flow problem is slightly different from that of the general method discussed before. The problem is also to minimize the total cost of the power generation. The problem is stated as:

\[ \text{Min} \left\{ \sum_{j=1}^{N} \left[ A_j P_j + B_j P_j + C_j (P_j)^2 \right] \right\} \]

\[ + \left[ A_N P_N + B_N P_N + C_N (P_N)^2 \right] \]  

s.t. \( P_k = g_1(V,\theta) \) for \( P,V \) and \( P,Q \) buses
\( Q_i = g_2(V,\theta) \) for \( P,Q \) buses

\[ \min \ P_j \leq P_j \leq \max \]  
\[ \min \ Q_j \leq Q_j \leq \max \]  
\[ \min \ V_k \leq V_k \leq \max \]  

where \( P_k, Q_i \) are the real and reactive power flow equations from \( [G(X,U)] = [0] \).

Instead of calculating the real power generation \( P_j, (j=1..N_g+1) \) directly, the voltage magnitudes and phase angles at all \( P,V \) and \( P,Q \) buses are computed. The objective function becomes a function of \( V \) and \( \theta \) if the real power balance equations at all the \( P,V \) buses are substituted into the objective function. Consequently, the cost function depends only on \( V \) and \( \theta \), and problem (d) becomes:

\[ \text{Min} \ f(V,\theta) = \text{Min} \ f(X) \]
\[ \text{s.t. } [G(V,\theta)] = [0] \]
\[ [H(V,\theta)] \leq [0] \]

Like the problem discussed before, this is a standard nonlinear programming problem. To solve the problem, various kinds of penalty function methods, such as the Ficco-McCormick, the Lootsma and the Zangwill methods [13], were attempted together with the DFP unconstrained minimization technique. The first two transformations require an initial state vector \([X]\) that does not produce inequality constraint violations, while the third one does not. Furthermore, because the Fiacco-McCormick and the Lootsma formulations require either the reciprocal or the logarithm of the equality and the inequality constraints as the penalty function (see appendix D), it is usually difficult to perform calculations with them. Therefore, the third method was preferred in [12]. With the Zangwill approach, the augmented objective function becomes:

\[
L(X,b^k,c^k) = f(X) + b^k \sum_{i=1}^{m} g^2_i(X) + \Sigma \text{ (penalties for violated inequality constraints)}
\]

(4.32)

Before the minimization of the cost function starts, a load flow problem is studied by using a nonlinear programming approach, such as the DFP method [13] or the Hessian matrix method [10]. It turns out that both methods cannot compete with the Newton-Raphson method so far as speed, accuracy and storage requirements, etc. are concerned.

When the initial value \([X]^0\) has been found, the necessary condi-
tion equations are solved. Because of dimensionality and sparsity considerations, none of the first order gradient methods, such as the DFP and the Conjugate Gradient, was adopted in [10]. Instead, the set of nonlinear, necessary condition equations are solved by a Newton-type method (using the Hessian matrix). Because of this, convergence with a specific set of penalty parameters is found to be very rapid. The initial value of the penalty parameters seem to be very important; and, also, the factors $b_k$ and $c_k$ cannot be chosen to be too big. As a result, overall convergence may be very long.

The main advantage of using the penalty function to handle the equality and/or inequality constraints is its simplicity. By adding the penalty function to the original objective function and selecting a proper penalty parameter, the constrained minimization problem simply becomes an unconstrained one. The techniques of solving the unconstrained minimization problem presented in the previous section can be effectively used.

Unfortunately, because the selection of the penalty parameter is nonsystematic [3], the implementation of the penalty function method exhibits slow convergence characteristics. Many researchers have tackled this problem. Some suggested that the use of the penalty function be kept to a minimum [7]. Some even developed other methods to avoid utilizing the penalty function. In the next section, the solutions to the problem (b) other than the penalty function method are presented. The flow chart of the algorithm using the Lagrange multiplier and the penalty function approach is given in Fig. 2.
Assume $[U]=[U]^0$

Get $[X]^k$ from $[G(X,U)]=[0], k=0,1...n$

Given penalty factors $b^k, c^k$

$$b^{k+1} = d b^k, \quad c^{k+1} = a c^k$$

$(d, a > 0), k=0,1...n$

Compute $[\lambda]$ at $[X]=[X]^k, [U]=[U]^k, c^k$

Compute gradient of $L(X,U,\lambda,c)$

$k=k+1$

Gradient of $L < \epsilon$?

Yes

Compute $[\delta U]^k$

Compute $[U]^{k+1} = [U]^k + [\delta U]^k$

$[U]^{\min} \leq [U] \leq [U]^{\max}$?

No

Set violated $u$'s to their upper or lower limit

Yes

Optimum reached, stop

No

Figure 2. OPF algorithm involving inequality constraints
4.1.2.2 Solutions of (b) Using the Indirect Method

The solutions of the nonlinear OPF Problem [b] without using the penalty function were found in references [17], [18].

There are basically two approaches. The first approach is the so-called the Generalized Reduced Gradient method, which is used in [18]. The main features of the algorithm are that whenever an improvement on [U] is made, the corresponding change in [X] due to the change in [U] is also computed using the sensitivity equation [29]

\[
\left[ \frac{\partial[G]}{\partial[X]} \right] \delta[X] + \left[ \frac{\partial[G]}{\partial[U]} \right] \delta[U] = [0] \tag{4.33}
\]

By keeping track of both changes in [X] and [U] (not to exceed their upper or lower limits), the whole computation process is carried forward. If \([X]^{k+1} = [X]^k + [\delta X]^k\) violates its limits, a change of basis of the control and the state vectors is performed. All the components of \([X]^k\) that violate the bounds are set at their corresponding bounds and are treated as control variables \([U]^{k+1}\) during the next iteration or iterations. Accordingly, an equal number of formerly control variables \([U]^k\) are changed into state variables \([X]^{k+1}\).

The main steps of the algorithm are:

1. Start with a feasible point \([U]^0\) and \([X]^0\).
2. Compute the gradient of \(L\) at \([X]=[X]^0\), \([U]=[U]^0\).
3. Compute the descent direction.
(4) The optimal step $B$ is next computed along the negative gradient direction of $f(X,U)$ by minimizing the scalar function

$$\min_B f[(X + B \delta X), (U + B \delta U)]$$

The advantages and disadvantages of the algorithm are:

(a) It is seen from step (3) that the step along the gradient is an optimal one. It is calculated based not only on the control vector $[U]$, but also on the state vector $[X]$ as well. This helps to avoid taking too large gradient steps, which sometimes results in the violations of more functional inequality constraints. When more inequality constraints are violated, more penalty terms must be added to the augmented objective function. This will definitely cause slow convergence.

(b) The careful movements in each iteration enable the method to reduce extra iterations caused by oscillations, especially about the optimal point. As a result, computer time is saved.

(c) The exchanges between $[U]$ and $[X]$ during the iteration process destroy the favorable properties of the standard power flow and sensitivity matrices. As a result, the computations of the Jacobian matrix and sparse matrix factorizations might become less sufficient and more time consuming. The method has been claimed to be successfully used to solved a practical power system (Alpha system) up to 30 buses. Due to the complexity of the exchange between $[U]$
and \([X]\) vectors during computations, the use of the algorithm to large scale power systems, say 500 buses or more, may be questionable [18].

In [17], Lee et al. uses a different approach called the Gradient Projection method [2]. This approach is effective for problems with nonlinear objective functions with linear constraints (both equality and inequality). In order for this optimization technique to be used, the ordinary nonlinear equality constraints (power flow equations) must be linearized with respect to the control vector \([U]\) at each iteration. After the linear equality constraint equation \([E][\delta U] = [F]\) of the power flow equations has been found, the projection matrix \([S]\) is calculated using

\[
[S] = I - [E]^t ([E] [E]^t)^{-1} [E]
\]  

(4.34)

where \([I]\) is an identity matrix and \([E], [F]\) are constant matrices at the kth iteration.

Finally, the adjustments of the control vector \(\delta U\) are set using

\[
[\delta U]^k = B^k [S]^k [g]^k
\]  

(4.35)

At the next iteration (\(k=k+1\)), control vector \([U]\) is

\[
[U]^{k+1} = [U]^k + [\delta U]^k
\]  

(4.36)

This seems to be a straightforward approach. Unfortunately, the performance of this algorithm is not given in [17].
4.2 OPTIMAL POWER FLOW BASED ON P-Q DECOMPOSITION

From the power system optimization point of view, the optimal Power Flow methods can be classified in one of the two categories-one providing exact solutions and the other approximate solutions. In this distinction, exact solutions take into account both real and reactive power flows in obtaining the solution; whereas, approximate methods achieve simplified representations and possibly computational efficiencies by ignoring either the real or reactive power flow equations. Approximate methods normally do not have the generality which exact methods possess.

The P-Q Decomposition Optimal Power Flow, based on the P-Q decoupled power flow, considers both the generality and accuracy attained by the exact methods and the simplicity and computational efficiency possessed by the approximate methods [14], [24].

The idea behind the P-Q decomposition power flow is that the real power flows are mostly sensitive to the bus voltage angles $\theta$ under power system steady state conditions. Likewise, the reactive power flows are mostly sensitive to the bus voltage magnitudes $V$. The coupling between the real power and phase angle (P-$\theta$) and the reactive power and bus voltage (Q-$V$) variables is relatively loose [9], [16]. Because of this, the exact Optimal Power Flow problem can be split into two subproblems, each of which is solved alone at a time during the computation process.
The objective function of the P-θ problem is still to minimize the total cost of power generation in a given system. The equality constraints are the bus real power balance equations. The control and the state variables of the Q-V problem are considered fixed during the P-θ problem solution; although the state variables may change slightly with each power flow solution, because of the actual coupling between P-θ and Q-V problems.

The control variables are:

\[
[U^P] = \begin{bmatrix}
\text{Real power generation P at all P,V bus} \\
\text{Phase shifter angles (if exist)}
\end{bmatrix}
\]

The state variables are:

\[
[X^P] = \begin{bmatrix}
\text{Bus voltage phase angle θ at all buses except that at the reference bus} \\
\text{Real power generation at the reference bus (functional form)} \\
\text{Real power flow in transmission lines (functional form)}
\end{bmatrix}
\]

The P-θ problem is stated as:

Minimize \( f(X^P,U^P) \) with respect to \([U^P]\)

s.t. \( [G^P(X^P,U^P)] = [0] \)

\[
[U^P]_{\text{min}} \leq [U^P] \leq [U^P]_{\text{max}}
\]

\[
[X^P]_{\text{min}} \leq [X^P] \leq [X^P]_{\text{max}}
\]
The objective function of the Q-V problem is selected to minimize the total transmission losses with the real power outputs $P_{Gj}$, $(j=1...N_g)$ at all the generation buses held constant. Since $P_{Gj}$ are all given, minimizing the real power output $P_{GN}(V)$ at the reference bus is identical with minimizing the total transmission losses.

The equality constraints are the reactive power balance equations. The control and the state variables of the P-Θ problem are considered fixed during the Q-V problem solution, although the state variables may change slightly with each power flow solution.

The control variables are defined as:

$$[u^q] = \begin{bmatrix}
\text{Voltage magnitudes V at all P,V buses} \\
\text{Tap values of tap-changing transformers (if exist)} \\
\text{Var injection values of shunt capacitors} \\
\text{or reactors (if exist)}
\end{bmatrix}$$

The state variables are:

$$[x^q] = \begin{bmatrix}
\text{Voltage magnitudes at P,Q buses} \\
\text{Var generation at P,V buses at the reference bus (function form)} \\
\text{Reactive power flows in transmission lines (function form)}
\end{bmatrix}$$

The Q-V problem is defined as:

$$\text{Min } F(x^q,u^q) = P_{GN}(x^q,u^q) \text{ with respect to } [u^q].$$
s.t. \[ G^q(X^q, U^q) = [0] \]
\[ [U^q]_{\text{min}} \leq [U^q] \leq [U^q]_{\text{max}} \]
\[ [X^q]_{\text{min}} \leq [X^q] \leq [X^q]_{\text{max}} \]

where: \( P_{GN} \) is the real power generation at the reference bus.

Having defined the two subproblems, one can solve them alternately using the solution techniques presented in the previous sections until the global optimum is reached. The algorithm is shown in Figure 3.
Figure 3. P-Q Decomposition Optimal Power Flow algorithm flow chart.
4.3 CONTINGENCY-CONSTRAINED OPTIMAL POWER FLOW

Up to this point in the presentation, it has been assumed that the constraints apply only to the power system in its intact condition. An extremely important extension to the problem is that the operation of the intact system should be optimized such that no limit violations occur after any defined contingency, such as outage of transmission lines, of generation apparatus, etc.

In power systems, particularly large scale power systems, a large number of contingency conditions can be expected. Consequently, the equality and the inequality constraints imposed by these contingencies are going to be enormous. Nevertheless, the number of binding contingent inequality constraints in the solution will normally be less than those imposed. Otherwise, from system operation considerations, the contingency constrained optimization may not be worth performing in the first place. With the contingency constraints involved in the Optimal Power Flow problem, the problem has become:

\[
\begin{align*}
\text{Min. } & f(X_0, U_0) \\
\text{s.t. } & [G_0(X_0, U_0)] = [0] \\
& [H_0(X_0, U_0)] \leq [0] \\
& [G_k(X_k, U_k)] = [0] \quad k=1,2\ldots s \\
& [H_k(X_k, U_k)] \leq [0] \quad k=1,2\ldots s
\end{align*}
\]

where: the subscripts 0's refer to the intact system.
k refers to each of the s contingency cases. 

\[ G(X_k, U_k) = [0] \] represents the power flow equations under the kth contingency case.

\[ H(X_k, U_k) \leq [0] \] is the inequality constraint vector equation under the kth contingent case.

The solution to the above problem is based primarily on solving the problem of the intact system (discussed before) as normal, testing the contingent inequalities at intervals, and adding to the problem some critical ones. At this stage the chosen contingent inequalities must be expressed through sensitivity analysis from \[ G(X_k, U_k) = [0] \] as functions of the intact system variables. The problem is that there may not be easily obtainable, differentiable relationships between the pre- and post-contingent variables. The problem can be approached through the control variables \([U_k]\). The post-contingency state \([X_k]\) can be completely defined by its control variables \([U_k]\), which may change from their pre-contingency values \([U_0]\) to their present values \([U_k]\) in non-simple ways in general. In order for the problem to be solvable, simplified assumptions have to be made. Usually, it is assumed that the control variables \([U_k]\) remain unchanged after contingency cases, that is:

\[ [U_0] = [U_k] \quad k=1 \ldots s \]  

(4.37)

Under this assumption, problem (g) becomes

\[
\begin{align*}
\text{Min} & \quad f(X_0, U_0) \\
\text{s.t.} & \quad [G_k(X_k, U_0)] = [0] \quad k=0,1,2 \ldots s \\
& \quad [H_k(X_k, U_0)] \leq [0] \quad k=0,1,2 \ldots s
\end{align*}
\]
The augmented objective function is formed similar to (4.23). The equality constraints are imposed again by the method of Lagrange multipliers. The inequality constraints are handled by the penalty function approach except for the control variables.

\[
L(X_0, X_1, X_s, U_0, \lambda_0, \lambda_1, \lambda_s) = f(X_0, U_0) + \sum_{k=0}^{s} \sum_{j=1}^{n} w_{kj}(X_k, U_0)
\]

\[
+ \sum_{k=0}^{s} \left( [\lambda_k]^t G_k(X_k, U_0) \right)
\]

\[
= f(X_0, U_0) + \sum_{k=0}^{s} P_k(X_k, U_0) + \sum_{k=0}^{s} \left( [\lambda_k]^t G_k(X_k, U_0) \right)
\]

(4.38)

where: \( P_k(X_k, U_0) = \sum_{j=1}^{n} w_{kj}(X_k, U_0) \)

With this formulation, the original problem has been changed into seeking the unconstrained minimization with respect to \([U_0]\).

For a stationary point of the augmented objective function, the following set of necessary conditions must be satisfied:

\[
\left[ \frac{\partial L}{\partial [X_0]} \right] = \left[ \frac{\partial f}{\partial [X_0]} \right] + \left[ \frac{\partial [G_0]}{\partial [X_0]} \right]^t [\lambda_0] + \left[ \frac{\partial P_0(X_0, U_0)}{\partial [X_0]} \right] = [0]
\]

(4.39)

\[
\left[ \frac{\partial L}{\partial [X_k]} \right] = \left[ \frac{\partial P_k}{\partial [X_k]} \right] + \left[ \frac{\partial [G_k]}{\partial [X_k]} \right]^t [\lambda_k] = [0] \quad k = 1, \ldots, s
\]

(4.40)

\[
\left[ \frac{\partial L}{\partial [U_0]} \right] = \left[ \frac{\partial f}{\partial [U_0]} \right] + \sum_{k=0}^{s} \left[ \frac{\partial [G_k]}{\partial [U_0]} \right]^t [X_k] + \sum_{k=0}^{s} \left[ \frac{\partial P_k}{\partial [U_0]} \right] = [0]
\]

(4.41)

\[
\left[ \frac{\partial L}{\partial [\lambda_k]} \right] = [G_k(X_k, U_0)] = [0] \quad k = 0, 1, \ldots, s
\]

(4.42)
The above sets of equations are little different in form from (4.2), (4.3) and (4.4) on page 19 for the base-case Optimal Power Flow. They can be solved using the same iteration technique with some extensions. It is easy to see that if no constraints are violated in the contingency cases \((s=0)\), the above equations reduce to the original Optimal Power Flow problem.

To solve the above equations, it is required, first of all, to solve the base-case problem for \(k=0\). The algorithm was presented in 4.1.1 and 4.1.2. Once the basic problem has been solved, the algorithm is extended by introducing the insecure contingency cases.

The extended algorithmic steps are:

(a) \(k=0\), solve the basic Optimal Power Flow problem.

(b) \(k=k+1\)

(c) Solve the fourth set of equations (4.42) for \([X_k]\) by the Newton method and check for any constraint violations. If there are no violations, go to step (g).

(d) Calculate the penalty terms for this case.

(e) Solve for \([\lambda_k]\) from (4.39) if \(k>0\), or from (4.40) if \(k=0\).

(f) Calculate the components of \(\frac{\partial L}{\partial [U_0]}\) based on (4.41) for the \(k\)th case, and add it to the existing \(\frac{\partial L}{\partial [U_0]}\).

(g) Go to step (b) unless \(k=s\).

(h) Check the elements of \(\frac{\partial L}{\partial [U_0]}\) for convergence. If converged, the optimum point has been reached. If not converged,
go to the next step.

(i) Perform a gradient step using \( \frac{\partial L}{\partial [U_0]} \) to obtain a new \([U]\) for the next iteration.

(j) Go to step (a).

It can be seen from the above computation procedure that the contingency-constrained Optimal Power Flow problem is much more complex than its basic problem.

The types of contingencies are typically the outage of one or more transmission lines, transformers, generators, static or synchronous compensators, bus loads, or combinations of these. In [19] the outages involving control variables is further examined. A linear relationship between control variables of the pre- and post- contingency is established through

\[
[U_k] = [M_k][U_0] \quad k=1,2,...,s \quad (4.43)
\]

instead of using the unchanged values of \([U_k]\) throughout the whole computation process. \([M_k]\) is normally a diagonal, constant matrix. Because \([U_k]\) is not equal to \([U_0]\), the third set of equations is not

\[
\left[ \frac{\partial L}{\partial [U_0]} \right], \text{ the required gradient vector, but } \left[ \frac{\partial L}{\partial [U_k]} \right] .
\]

To calculate the components of \( \frac{\partial L}{\partial [U_0]} \) due to outage case \( k \), the transformation

\[
\left[ \frac{\partial L}{\partial [U_0]} \right] = \left[ \frac{\partial [U_k]}{\partial [U_0]} \right] \left[ \frac{\partial L}{\partial [U_k]} \right] = [N^k] \left[ \frac{\partial L}{\partial [U_k]} \right] \quad (4.44)
\]
must be invoked. Other steps are the same as before.

Numerical results on an IEEE 30-bus standard load flow test system are given in [19], the test results show that from the normal system operation point of view, the Optimal Power Flow with steady state security constraints can only be considered to be suboptimal because more savings are possible if contingency constraints are not considered. On the other hand, as far as security is concerned, if outages happen on some transmission lines, some quantities do exceed their limits, which may not be acceptable.

Contingency constraint sets are additional constraints in the Optimal Power Flow problems. Some researchers stressed that they must be considered, others thought that there was no need to consider them at least there was no need to consider many of them because in power system steady state operations, the probability of outages of transmission lines and other power system apparatus is small. Therefore, the fact should be aware of and taken advantage of in order to get the best economic benefit through the study of OPF.

In short, the number of additional constraints which should be added to the Optimal Power Flow problem is still an open question, and it depends on a lot factors. The coordination between the power system economic operation and reliability should be considered simultaneously.
4.4 THE OPTIMAL POWER FLOW SOLUTION USING QUADRATIC PROGRAMMING

The Optimal Power Flow problem, especially the economic dispatch problem has been formulated as a quadratic programming problem [25], [26] and solved using two kinds of quadratic programming approaches [27], [28].

Quadratic programming is, by definition, the problem of finding the minimum (or maximum) value of a quadratic function subject to equality and/or inequality constraints.

Since the objective function of the total cost in terms of the real power outputs of all the generators in a given power system is assumed to be quadratic in the economic dispatch problem, it might be possible to use the quadratic programming approach to solve the economic dispatch problem.

In [25], the economic dispatch problem is solved using the Wolfe quadratic programming technique [27] which solves the following quadratic function

\[
\begin{align*}
\text{Min} & \quad (d + [C]^t [X] + 0.5 [X]^t [Q] [X]) \\
\text{s.t.} & \quad [A] [X] = [B] \\
& \quad [X] \geq [0]
\end{align*}
\]

where: \(d\) is a scalar constant.

[C] stands for an n by 1 vector of constants.

[Q] is an n by n semidefinite square matrix.

[A] is an m by m matrix.
[B] is an m by 1 vector.

[X] represents an m by 1 vector of unknowns.

In (i), the condition \([X] \geq [0]\) is an additional constraint imposed by the Wolfe method. The removal of the equality constraints is achieved by introducing a vector \([\lambda]\) of Lagrange multipliers.

\[
F(X) = d + [C]^t[X] + 0.5 [X]^t[Q] [X] + [\lambda]^t([A][X] - [B]) \tag{4.45}
\]

Wolfe proves in [27] that if \([X]\) can be found such that \([A][X] = [B]\) and there exist a \([V] \geq [0]\) (\([V]\) is an n by 1 vector), and \([\lambda]\) (an m by 1 vector) such that

\[
[V]^t [X] = [0] \tag{4.46}
\]

\[
[Q][X] - [V] + [A]^t [\lambda] + [C] = [0] \tag{4.47}
\]

\[
[V] \geq [0] \tag{4.48}
\]

then \([X]\) minimizes the quadratic objective function.

Equations (4.46)-(4.48) give the necessary conditions for optimality. It is seen that, since both \([X]\) and \([V]\) are restricted to being equal to or greater than zero, the only way that equation (4.46) is satisfied is if \([X] > 0\), then \([V] = [0]\) or vice versa. As a result, equations (4.46)-(4.48) are linear equations in \([X]\), \([V]\) and \([\lambda]\). Thus the quadratic programming problem is reduced to the problem of solving a set of linear equations. In the Wolfe method, equations (4.46)-(4.48) are solved by employing the simplex method of linear programming [2], [3].

If inequality constraints are involved in the minimization problem, which is usually the case in the economic dispatch problem,
slack variables must be introduced to transfer the inequality constraints into equality constraints.

It should be noted from the above introduction to the Wolfe algorithm that this approach requires a quadratic objective function and a system of linear equality constraints. It also imposes the restriction that all system variables are non-negative. The economic dispatch problem, as normally defined, does not meet the first and the second requirements for the following reasons:

(1) Although the cost function of the real power generation at the reference bus is quadratic, it is an unknown variable and cannot be treated as a control. It is usually necessary that it be substituted by other control and state variables. This, in general, results in a nonlinear cost function rather than a purely quadratic one.

(2) The equality constraints are nonlinear equations.

In [25], in order to overcome these difficulties, several steps are taken.

(a) The real power outputs of all the generators including that of the reference bus, are substituted by the real balance equations in terms of real and imaginary bus voltages (rectangular coordinates are used). The resultant cost function is then approximated quadratically.

(b) Real power equality constraint equations $P=f(e,f)$ are linearized with respect to the real part of $e$ and imaginary part $f$ of bus voltage.
(c) Inequality constraints are handled by introducing slack variables.

With these modifications, the first and the second requirements for using the Wolfe quadratic programming technique have now been met. Unfortunately, because some new variables in the cost function (imaginary parts of bus voltages \( f \)) may be negative, the third requirement cannot yet be met at this point. In [25], the positivity of all variables is insured through shifting coordinates. This method is based on the knowledge that the magnitude of the unrestricted variables cannot exceed certain values. When this is known, the coordinate system can be shifted. The new system variables are related to the original variables by constants.

\[
\begin{align*}
x_{iN} &= x_{i0} - |x_{i0}|_{\text{max}} \\
\end{align*}
\]

where:
\( x_{iN} \) = new restricted variable.
\( x_{i0} \) = original unrestricted variable.
\( |x_{i0}|_{\text{max}} \) = a constant that \( x_{i0} \) can attain.

Up to this point, the three requirements that the Wolfe algorithm requires are all met. The method can now be used to solve the economic dispatch problem. The results of computation on several systems of different sizes (5 bus, IEEE 14, 30, 57, 118 bus systems) are given in [25]. Convergence was obtained in three iterations for all test systems and the solution time is claimed to be short enough to allow the method to be used for on-line dispatching problems. However, some people doubted the possibility of utilizing the method for on-line applications (see [25] discussion). According to their
opinion, both the core storage used and the computer solution time required appeared to be very high.

The other quadratic programming approach can be found in [26]. This approach is different from that of the previous one in the following aspects:

1. Real power outputs are solved for directly instead of voltage.
2. The Beale quadratic programming method [28] is adopted instead of Wolfe's.
3. Real power balance is considered as a whole just as in the classical economic dispatch problem.
   \[ P_L - P_G + P_{TL} = 0 \]
4. Reactive power dispatch problem is also solved together with the economic dispatch problem using gradient method.
5. The load prediction problem was included, which is normally not considered in Optimal Power Flow problems.
6. Power Flow studies were performed using the Gauss-Seidel method.

The complete optimization procedure for the solution of the real and reactive power dispatch problem is outlined in [26] and summarized below:

(1) Input system data.
(2) Calculate the system impedance matrix [Z].
(3) Perform Power Flow using Gauss-Seidel method.
(4) Calculate transmission losses.
(5) Calculate total generation required (Load + Losses).
(6) Solve quadratic programming problem for real power dispatch.
(7) Perform another Power Flow with the new real power dispatch.
(8) Calculate transmission losses.
(9) Test if losses have converged. If not, go to step (5).
(10) Determine system production cost.
(11) Is a reactive power dispatch required? If not, go to step (15).
(12) If a previous reactive dispatch has been made, has the system production cost converged? If so, go to step (15).
(13) Allocate reactive power using gradient method.
(14) Calculate transmission losses and go to step (5).
(15) Output computed data. Stop.

It can be observed that the above procedure seems to be too complex to be used for on-line applications. The computational results for a 10-bus, 22-generator network are given in [26]. The computing time required for the full real and reactive solution is estimated to be less than one minute on a 1907 computer which has a cycle time of 2.1 μ second. For large scale power systems, computer time is expected to be very high. The storage requirements are also high according to [26]; therefore, a relatively large computer, had to be used for maintaining computation speed and accuracy.
SECTION 5. A SUMMARY OF THE CHARACTERISTICS OF THE VARIOUS OF METHODS

The characteristics of the various of methods are summarized in this section. These characteristics include computational speed, reliability, generality and simplicity etc. The necessity for accuracy varies according to application. In general, accuracy cannot be expected to be too high because of the high dimensionality of the OPF problem. The potential for the OPF solution to be used for on-line applications will probably be best achieved with reduced accuracy as well.

In the vast majority of methods, the minimization process is iterated through a power flow solution by the Newton-Raphson method. This method is fast, accurate and reliable. It is also seen that the Newton-Raphson approach is particularly well-suited in calculating the Lagrangian multipliers; since the equations for optimality are linear in $\lambda$.

The decoupled power flow algorithm is also a widely adopted method for calculating the OPF problem. It is handy to use this algorithm in P-Q decomposition OPF studies. The Lagrangian multipliers are computed iteratively [16].

It is easy to see that the Dommel-Tinney approach is the most general and straightforward Optimal Power Flow approach. The Optimal
Power Flow formulation based on Dommel-Tinney is exact. There are a lot of other papers which are also based on the exact formulation [9], [16] etc. With this kind of problem formulation, the power system optimization problem can be solved precisely. In the Dommel-Tinney approach, the changes in \([U]\) are obtained using the Steepest Descent method. The method usually works well in the first few iterations. Numerical difficulties are encountered about the optimum, particularly when penalty functions are involved. A better convergence property can be expected if other methods, such as the Conjugate Gradient, the Newton and the DFP, are adopted. These methods have been successfully utilized by many researchers [10], [11], [13], [15], [16], etc. The basic storage requirements for the Dommel-Tinney type of methods are the upper and lower triangularized, sparse Jacobian matrices, Lagrangian multiplier vector \([\lambda]\), and other \(n\) by 1 vectors.

In [10]-[13], the formulation of the Optimal Power Flow problem is also exact. By substituting the real power generation of all the \(P,V\) buses and using the real power generation equations, the \(P\) Optimal Power Flow problem is changed to optimally finding the voltage magnitudes and phase angles at all the \(P,V\) buses. As a result, the equality constraints are reduced from \((N-N_g)\) to \(2N_L\) (in this method, no reference bus is assigned). This may lead to the reduction of storage requirements in the power flow study. On the other hand, the elimination of the power equations of the generator buses from the power flow equations through substitution for \(P_j\), \((j=1...N_g+1)\) to the objective function causes the number of control
variables to double. Another feature of the approach is that the penalty function is employed to both equality and the inequality constraints. Consequently, numerical difficulties might appear if large power systems are studied using this method. In [13], the storage requirements are a $2N_g$ by $2N_g$ symmetrical full matrix, several $2N_g$ vectors required by the DFP method plus the penalty weighting vectors [B] and [C]. As far as the storage requirement of [10] is concerned, symmetrical, triangularized sparse Jacobian and Hessian matrices and [B], [C] matrices are needed to be stored.

The Optimal Power Flow approach based on P-Q decomposition is a well-approximated method compared with the previous methods. This two-stage approach splits control variables in two parts and handles one part at a time. The storage requirement of this method is no more than that of the above methods, with computational efficiency enhanced.

The OPF problem involving system security requires considerably more computer time and storage requirements than those of the problem without the security constraints, depending on how many contingency constraints are imposed. On the other hand, by introducing security constraints, a power system can operate both economically and safely. This is vital in large-scale modern power systems.

Solving the Optimal Power Flow problem using quadratic programming approaches requires that a quadratic objective function and linear equality and/or inequality constraints be available. Because of this, the Taylor series expansion technique is often employed.
The main advantage of using this approach is the avoidance of penalty functions in handling the inequality constraints.
SECTION 6. CONCLUSIONS/IMPLEMENTATION OF OPTIMAL POWER FLOW AT OSU

The Optimal Power Flow problem is an optimization problem involving a large number of variables. Through the study of the Optimal Power Flow using modern mathematical programming techniques, the following can be obtained:

1. Economic operation of power systems.
2. Better system voltage profile.
3. Power system static security (if security constraints are imposed).

The Optimal Power Flow solutions using nonlinear and quadratic programming techniques are now available, and many optimization methods have been used. It is seen that the choice of optimization methods is largely dependent on the objectives, controls and constraints considered. For on-line applications, the mathematical models used are usually simplified to meet the needs of speed and reduced storage requirements at the expense of accuracy. The Optimal Power Flow solutions using piecewise cost function and linear programming technique are also available. A review is given in [21].

Further development of OPF depends partially on the development of mathematical programming techniques. However, the development in other fields, such as large-scale system theory, computers, etc. will also be important.
The implementation of Optimal Power Flow requires three basic computer program subroutines. They are:

(1) A Newton-Raphson power flow program or a fast decoupled Newton power flow program.

(2) An optimization subroutine based on, for example, the Conjugate Gradient, or the DFP algorithm.

(3) A subroutine for solving large-scale, sparse, linear equations using a Gaussian elimination algorithm or other effective algorithm.

The three subroutines are currently available at OSU. The latest version of the Newton power flow program which can solve power flow problems up to 1500 buses can be obtained from BPA. With this power flow program, the OPF program will be able to solve extremely large-scale OPF problems.
BIBLIOGRAPHY


[21] B. Stott, J.L. Marinho and O. Alsac, "Review of Linear Program-


APPENDICES
APPENDIX A

Economic Dispatch Considering Losses

In the classic economic dispatch problem, the objective function \( f \) is taken to minimize the total power generation cost subject to the equality constraint of total real power balance equation (inequality constraints are disregarded at the moment). Suppose there are \( N_g + 1 \) generators in a given system.

\[
\text{Min } f = \sum_{j=1}^{N_g+1} k(P_j) = \sum_{j=1}^{N_g+1} (A_j + B_j P_j + C_j(P_j)^2)
\]

s.t. \( P_G - P_L - P_{TL} = \sum_{j=1}^{N_g+1} P_{Gj} - P_L - P_{TL} = 0 \)  \hspace{1cm} (A.1)

where: \( P_G = \sum_{j=1}^{N_g+1} P_{Gj} \) is the total real power generation by \( N_g + 1 \) generators,

\( P_L \) stands for the total load,

\( P_{TL} \) is the total transmission losses.

Using the Lagrangian multiplier approach as was used before, one formulates the augmented objective function

\[
L(P_{Gj}, \lambda) = \sum_{j=1}^{N_g+1} (A_j + B_j P_{Gj} + C_j(P_{Gj})^2) + \lambda (P_G - P_L - P_{TL})
\]

At the minimum, \( \frac{\partial L}{\partial P_{Gj}} = 0 \) \hspace{1cm} j=1...N_g+1
or
\[ B_j + 2C_j P_{Gj} + \lambda \left( 1 - \frac{\partial P_{TL}}{\partial P_{Gj}} \right) = 0 \]  \hspace{1cm} (A.3)

\[ \lambda = \frac{B_j + 2C_j P_{Gj}}{1 - \frac{\partial P_{TL}}{\partial P_{Gj}}} = \lambda_j \quad j=1...N_g+1 \]  \hspace{1cm} (A.4)

In the above computations, \( P_L \) is considered to be fixed. \( P_{TL} \) is unknown and depends on the real power generation of each generator. In order to calculate the Lagrangian multipliers \( \lambda_j = \lambda \) (called incremental operating cost), the partial derivatives of \( P_{TL} \) with respect to each of the power generator \( P_{Gj} \) must be evaluated.

Usually, the total system loss is assumed to be a quadratic function of the \( P_{Gj} \) 's:

\[ P_{TL} = g \sum_{i=1}^{N_g+1} \sum_{j=1}^{n} B_{ij} P_{Gi} P_{Gj} \]  \hspace{1cm} (A.5)

\[ = \{P_G\}^t [B] [P_G] \]  \hspace{1cm} (A.6)

where: \( \{P_G\}^t = [P_{G1} \ P_{G2} ... P_{Gn} \ P_{G_{N+1}}] \)

\( [B] = \) constant \( n \) by \( n \) symmetrical matrix with general entry \( B_{ij} = B_{ji} \) (\( i \neq j \)).

The remaining problem is to compute the \( B \) matrix parameters \( B_{ij} \), which can be implemented by several techniques. One approach, called the Hill-Stevenson method [30], calculates \( B \) constants using partial derivatives. A description of the method follows.

Notice that:
\[ \frac{\partial P_{TL}}{\partial P_{Gi}} = 2 g \sum_{j=1}^{N_g+1} B_{ij} P_{Gj} \]  \hspace{1cm} (A.7)
\[ \frac{\partial^2 P_{TL}}{\partial P_{Gi} \partial P_{Gj}} = 2 B_{ij} \]  
(A.8)

\[ B_{ij} = \frac{1}{2} \left[ \frac{\partial^2 P_{TL}}{\partial P_{Gi} \partial P_{Gj}} \right] \]  
(A.9)

The total power injected into the transmission lines $P_{TJ}$ is:

\[ P_{TJ} = R_e \left[ \sum_{i=1}^{n} V_i (I_i)^* \right] \]
\[ = \sum_{i=1}^{n} \sum_{j=1}^{n} V_i V_j \left[ G_{ij} \cos(\theta_i - \theta_j) + B_{ij} \sin(\theta_i - \theta_j) \right] \]  
(A.10)

In general, $P_{TJ}$ would equal the total transmission system loss $P_{TL}$

\[ P_{TJ} = P_{TL} \]  
(A.11)

Therefore,

\[ \frac{\partial P_{TL}}{\partial \theta_j} = 2 \sum_{i=1}^{n} V_i V_j G_{ij} \sin(\theta_i - \theta_j) \]  
(A.12)

\[ \frac{\partial P_{TL}}{\partial P_{Gj}} = \sum_{k=1}^{n} \frac{\partial P_{TL}}{\partial \theta_k} \frac{\partial \theta_k}{\partial P_{Gj}} \]  
(A.13)

The partial derivatives $\frac{\partial \theta_k}{\partial P_{Gj}}$ are computed as follows:

First, perform a load study to get $\theta_k$ (k=1...n) and solve the lossless economic dispatch problem as a first approximation.

Second, increase all bus loads proportionally by a small amount (10% for example) and allow the jth generator (j=1...Ng+1) to pick
up the increased load alone.

Third, perform another load flow study to solve for the phase angle $\Theta_k$ at all buses ($k=1...n$), from which the change in phase angle $\Theta_k$, ($k=1...n$) can be evaluated. Therefore,

$$\frac{\partial \Theta_k}{\partial P_{Gj}} = \frac{\partial \Theta_k}{P_{Gj}} = A_{kj} = \text{constant}$$

$$k=1,2,...n$$

$$j=1,2,...n$$

(A.14)

The second partial derivatives of $P_{TL}$ with respect to $P_{Gi}$ is computed next.

From (A.14), we have:

$$\frac{\partial^2 P_{TL}}{\partial P_{Gi} \partial P_{Gj}} = \frac{\partial}{\partial P_{Gi}} \left[ \sum_{k=1}^{n} \frac{\partial P_{TL}}{\partial \Theta_k} \left( \frac{\partial \Theta_k}{\partial P_{Gj}} \right) \right]$$

$$j=1...N+1$$

(A.15)

$$= \sum_{m=1}^{n} \frac{\partial}{\partial \Theta_m} \left[ \sum_{k=1}^{n} \frac{\partial P_{TL}}{\partial \Theta_k} \left( \frac{\partial \Theta_k}{\partial P_{Gj}} \right) \right] \left( \frac{\partial \Theta_m}{\partial P_{Gi}} \right)$$

(A.16)

Therefore,

$$\frac{\partial^2 P_{TL}}{\partial P_{Gi} \partial P_{Gj}} = \sum_{m=1}^{n} \sum_{k=1}^{n} \frac{\partial^2 P_{TL}}{\partial \Theta_m \partial \Theta_k} A_{kj} A_{mi}$$

(A.17)

where: $A_{kj} = \frac{\partial \Theta_k}{\partial P_{Gj}}$, $A_{mi} = \frac{\partial \Theta_k}{\partial P_{Gi}}$

From (A.13),

$$\frac{\partial^2 P_{TL}}{\partial \Theta_m \partial \Theta_k} = \begin{cases} 
2 V_m V_k G_{mk} \cos(\Theta_m - \Theta_k) & m \neq k \\
-2 \sum_{i=1}^{n} V_i V_m G_{im} \cos(\Theta_i - \Theta_m) & m = k
\end{cases}$$

(A.18)
Finally, 

\[ B_{ij} = \frac{1}{2} \sum_{m=1}^{n} \sum_{k=1}^{n} \frac{\partial^2 p_{TL}}{\partial \theta_m \partial \theta_k} A_{mi} A_{kj} \]  

(A.19)

\[ \lambda_j = \lambda = \frac{B_{ij} + 2 C_j P_{Gj}}{N + 1} \quad j=1...N+1 \]  

(A.20)

If the inequality constraints of real power generation of each generator are included, the problem becomes:

\[ \text{Min} \quad f = \min_{j=1}^{N+1} K(P_j) \]  

s.t. \[ P_G - P_L - P_{TL} = 0 \]  

\[ \min \quad \max \quad P_{Gj} \leq P_{Gj} \leq P_{Gj} \quad j=1...N+1 \]  

(A.21)

In this case, in the augmented objective function, the so-called Kuhn-Tucker dual variables must be present.
APPENDIX B

Differential Operations of multivariable Functions

In the optimization problems, the following three kinds of operations are usually encountered.

(a) Differentiation of a scalar function with respect to a vector

Let \( f(X) \) be a scalar function of \( m \) variables \( x_i, i=1...m \), \( [X]=[x_1 \ x_2...x_m]^t \). The derivative of \( f(X) \) with respect to vector \( [X] \) is defined as:

\[
\left[ \frac{df(X)}{d[X]} \right] = \left[ \frac{\partial f}{\partial x_1} \ \frac{\partial f}{\partial x_2} \ \ldots \ \frac{\partial f}{\partial x_m} \right]^t
\]  (B.1)

\[
\left[ \frac{df(X)}{d[X]} \right]
\]

is called the gradient of \( f(X) \).

If \( f \) is a function of two vectors \( [X], [U] \), \( f=f(X,U) \)

where : \( [X]=[x_1 \ x_2...x_m]^t \)
\( [U]=[u_1 \ u_2...u_p]^t \)

then

\[
\left[ \frac{\partial f(X)}{\partial [X]} \right] = \left[ \frac{\partial f(X,U)}{\partial x_1} \ \frac{\partial f(X,U)}{\partial x_2} \ \ldots \ \frac{\partial f(X,U)}{\partial x_m} \right]^t
\]  (B.2)

\[
\left[ \frac{\partial f(X)}{\partial [U]} \right] = \left[ \frac{\partial f(X,U)}{\partial u_1} \ \frac{\partial f(X,U)}{\partial u_2} \ \ldots \ \frac{\partial f(X,U)}{\partial u_p} \right]^t
\]  (B.3)
(b) Differentiation of a vector function with respect to a vector

Let \([\mathbf{F}(\mathbf{X})]\) be a vector function,

\[
[\mathbf{F}(\mathbf{X})] = \begin{bmatrix} f_1(\mathbf{X}) & f_2(\mathbf{X}) & f_3(\mathbf{X}) & \cdots & f_n(\mathbf{X}) \end{bmatrix}^t
\]

\([\mathbf{X}] = \begin{bmatrix} x_1 & x_2 & \cdots & x_m \end{bmatrix}^t
\]

The derivative of the vector function \([\mathbf{F}(\mathbf{X})]\) with respect to vector \([\mathbf{X}]\) is defined as:

\[
\begin{bmatrix}
\frac{\partial [\mathbf{F}(\mathbf{X})]}{\partial [\mathbf{X}]^t}
\end{bmatrix} = \begin{bmatrix}
\frac{\partial f_i}{\partial x_j}
\end{bmatrix}_{nm} = \begin{bmatrix}
\mathbf{J}
\end{bmatrix}
\]

\((i=1\ldots n, j=1\ldots m)
\]

\([\mathbf{J}]\) is called the Jacobian matrix.

(c) The second derivative of a scalar function \(f(\mathbf{X})\) with respect to a vector

The second derivative of \(f(\mathbf{X})\) is defined as:

\[
\begin{bmatrix}
\frac{\partial^2 f(\mathbf{X})}{\partial ([\mathbf{X}]^t)^2}
\end{bmatrix} = \begin{bmatrix}
\frac{\partial f(\mathbf{X})}{\partial x_i}\frac{\partial f(\mathbf{X})}{\partial x_j}
\end{bmatrix}_{mm} = \begin{bmatrix}
\mathbf{A}
\end{bmatrix}
\]

\((i=1\ldots m, j=1\ldots m)
\]

In (B.5), \([\mathbf{A}]\) is called the Hessian matrix. The positive definite of \([\mathbf{A}]\) at \([\mathbf{X}]=[\mathbf{X}^*]\) implies that \([\mathbf{X}^*]\) is the local minimum of \(f(\mathbf{X})\).
APPENDIX C

The Kuhn-Tucker Theorem

The minimization problem with equality constraints can be solved by utilizing Lagrange multipliers. Given the following problem:

\[ \text{Min } f(X) \]
\[ \text{s.t. } [G(X)] = [0] \]  \hspace{1cm} (C.1)

where: \( f(X) \) is a scalar function

\[ [X] = [x_1 \ x_2 \ ... \ x_m]^t \]
\[ [G(X)] = [g_1(X) \ g_2(X) \ ... \ g_n(X)]^t \]

The augmented Lagrange function is defined as:

\[ L(X, \lambda) = f(X) + [\lambda]^t [G(X)] \] \hspace{1cm} (C.2)

or:

\[ L(X, \lambda) = f(X) + \sum_{i=1}^{n} \lambda_i g_i(X), \] \hspace{1cm} (C.3)

\( \lambda_i, i=1,2,...n \) are called Lagrange multipliers. They are introduced to simplify the computations.

The necessary conditions for local minimum are:

\[ \left[ \begin{array}{c} \frac{\partial L}{\partial [X]} \\ \frac{\partial L}{\partial [\lambda]} \end{array} \right] = \left[ \begin{array}{c} \frac{\partial f(X)}{\partial [X]} \\ \frac{\partial [G(X)]}{\partial [X]} \end{array} \right] + [\lambda]^t \left[ \begin{array}{c} \frac{\partial [G(X)]}{\partial [X]} \end{array} \right] = [0] \] \hspace{1cm} (C.4)

\[ \left[ \begin{array}{c} \frac{\partial L}{\partial [\lambda]} \end{array} \right] = [G(X)] = [0] \] \hspace{1cm} (C.5)
The above equations can also be written in another form:

\[
\frac{\partial L}{\partial x_j} = \frac{\partial f(X)}{\partial x_j} + \sum_{i=1}^{n} \lambda_i \frac{\partial g_i}{\partial x_j} \quad j=1...m \quad (C.6)
\]

\[
\frac{\partial L}{\partial \lambda_i} = g_i(X) = 0 \quad i=1...n \quad (C.7)
\]

The unknown variables in the above equations are \(x_j\) (\(j=1...m\)) and \(\lambda_i\) (\(i=1...n\)). Notice that the total number of unknowns \((m+n)\) is equal to that of the above equations. Therefore, \(x_j\) and \(\lambda_i\) can be uniquely determined to get the local minimum point.

The Kuhn-Tucker theorem is the extension of the above method. It gives the necessary conditions for the above problem with inequality constraints involved. Consider the following problem:

\[
\text{Min } f(X) \\
\text{s.t. } g_i(X) \leq 0 \quad i=1...n \quad (C.8)
\]

Define:

\[
L(X,\lambda) = f(X) + \sum_{i=1}^{n} \lambda_i g_i(X) \quad (C.9)
\]

According to the Kuhn-Tucker theorem, at the minimum, the following conditions must be satisfied:

\[
\frac{\partial L}{\partial x_j} = \frac{\partial f}{\partial x_j} + \sum_{i=1}^{n} \lambda_i \frac{\partial g_i}{\partial x_j} = 0 \quad j=1...m \quad (C.10)
\]

\[
\begin{cases}
\lambda_i = 0 & \text{if } g_i(x) < 0 \quad i=1...n \quad (C.11) \\
\lambda_i \geq 0 & \text{if } g_i(x) = 0 \quad i=1...n \quad (C.12)
\end{cases}
\]
In (C.9)-(C.12), $\lambda_i$ are called Kuhn-Tucker multipliers. They are similar to Lagrange multipliers, the difference being that the sign of the Kuhn-Tucker multipliers are specified as $\lambda_i \geq 0$, while the Lagrangian multipliers can be either positive or negative.

The above theorem says that if the local minimum point $[X]=[X^*]$ occurs, with all the following conditions $g_i(X^*)<0$ satisfied, $\lambda_i=0$. This is exactly an unconstrained minimization problem. If the local minimum point $[X]=[X^*]$ satisfies $g_i(X^*)=0$, then $\lambda_i \geq 0$. This is an equality constrained minimization problem. There exists another case in which some components of the vector $[X]$ satisfy $g_i(x_j^*)=0$, for $j=1...r$ $(r<m)$, some satisfy $g_i(x_j^*)<0$, for $j=r+1, r+2,...m$. For the first case, the corresponding $\lambda_j$ is greater than zero, the second case $k_j$ equal to zero. In short, we always have enough information to find the local minimum point, while all the constraints are satisfied.
APPENDIX D

The Penalty Function Method

1. solving optimization problems under equality constraints using the penalty function method

The penalty function method is widely used in dealing with constrained optimization problems. The basic idea of the method is to change the original constrained problem into solving a sequential unconstrained minimization problems by properly choosing a so-called weighting factor. The method is also called "Sequential Unconstrained Minimization Technique ", or "SUMT ".

Let the optimization problem be:

\[
\begin{align*}
\text{Min} & \quad f(X) \\
\text{s.t.} & \quad g_i(X) = 0 \quad i=1,2,...n
\end{align*}
\]  

(D.1)

The augmented objective function is formed as follows:

\[
P(X,b^k) = f(X) + b^k \sum_{i=1}^{n} [g_i(X)]^2
\]  

(D.2)

The way in which the augmented objective function in constructed is based on the Zangwill "Exterior SUMT" method (see the last part of this appendix).

\[b^k, (k=1...n)\] is called the weighting factor. It is numerically given an initial value at the beginning of the computation and increased during the numerical solution process. Whenever \( g_i(X) = 0 \),
the value of \( P(X, b^k) \) is equal to that of \( f(X) \). On the other hand, if 
\( g_i(X) \) is not zero, additional value is added to \( f(X) \).

\[
b^k \sum_{i=1}^{n} [g_i(X)]^2
\]

is called the penalty term. If for different \( g_i(X) \), different weighing factors \( b_i^k \) are added, the penalty term can be written as

\[
\sum_{i=1}^{n} b_i^k [g_i(X)]^2
\]  

(D.3)

In the above expression, \( i \) represents the \( i \)th weighting factor associated with the \( i \)th equality constraint \((g_i(X)=0)\), \( k \) represents the \( k \)th iteration. When the penalty term is added to the original objective function, the constraints are released. By choosing different values of \( b^k \), usually \( b^{k+1} > b^k \), a sequence of unconstrained minimization problems can be obtained, each of which is going to be minimized. Iteration procedure is presented below:

1. Given an initial value of \( b^k, k=0 \)
2. Minimize the augmented objective function using any unconstrained minimization technique such as the Fletcher-Powell, the Conjugate Gradient etc. to get the minimum point \([X]^k\) for \( b^k, k=0 \).
3. Check whether all the equality constraints are satisfied. If yes, stop, if not, increase \( b^k, k=k+1 \), go to step 2.

2. Comparison between Lagrangian multiplier method and the penalty function method.

Suppose that the original minimization problem is

\[
\text{Min} \quad f(X)
\]
s.t. \( g_i(X) = 0 \quad i=1...n \) \hspace{1cm} (D.4)

In the Lagrangian multiplier method, the augmented objective function is:

\[
L(X,\lambda) = f(X) + \sum_{i=1}^{n} \lambda_i g_i(X)
\] \hspace{1cm} (D.5)

In the penalty function method, the augmented objective function is:

\[
P(X,b^k) = f(X) + \sum_{i=1}^{n} b_i^k [g_i(X)]^2
\] \hspace{1cm} (D.6)

Differentiating (D.5) with respect to \( x_j \), (\( j=1...m \)), we have:

\[
\frac{\partial L(X,\lambda)}{\partial x_j} = \frac{\partial f}{\partial x_j} + \sum_{i=1}^{n} \lambda_i \frac{\partial g_i(X)}{\partial x_j} = 0 \quad j=1...m \] \hspace{1cm} (D.7)

Differentiating (D.6) with respect to \( x_j \), (\( j=1...m \)), we have:

\[
\frac{\partial P(X,b^k)}{\partial x_j} = \frac{\partial f}{\partial x_j} + 2 \sum_{i=1}^{n} g_i(X) b_i^k \frac{\partial g_i(X)}{\partial x_j} = 0 \quad j=1...m \] \hspace{1cm} (D.8)

Comparing (C.7) with (C.8), we derive:

\[
\frac{\lambda_i}{\Sigma_{i=1}^{n} \lambda_i} \frac{\partial g_i(X)}{\partial x_j} = 2 \sum_{i=1}^{n} [b_i^k g_i(X)] \frac{\partial g_i(X)}{\partial x_j}
\] \hspace{1cm} (D.9)

or:

\[
g_i(X) = \frac{\lambda_i}{2b_i^k} \quad i=1...n \] \hspace{1cm} (D.10)

The above expression clearly indicates the relationship between the Lagrangian multiplier \( \lambda_i \) and the penalty weighting factor \( b_i^k \).
When \( \lambda_i \) is equal to zero we have \( g_i(X) \) is zero. This means the equality constraints are satisfied. On the other hand, when \( b_i \) approaches infinity, we also have \( g_i(X) = 0 \), which means the equality constraints are satisfied.

3. Solving the minimization problem under equality and inequality constraints

The optimization problem now becomes:

\[
\begin{align*}
\text{Min} & \quad f(X) \\
\text{s.t.} & \quad g_i(X) \geq 0 \quad i=1...n
\end{align*}
\]

(D.11)

In this case, the minimization point for \( f(X) \) can occur either on the feasible region or in it.

Iteration can be started either in the feasible region or out of it. For the former case, the iteration is performed in the region at all times until the minimization point is found. The method is called "Interior SUMT". For the latter case, iteration commences outside the feasible region. During the iteration procedure, \([X]\) moves into the feasible region from outside. This method is called "Exterior SUMT".

For the two methods mentioned above, the penalty functions added to the original objective function are different.

In the "Interior SUMT" method, the penalty function is defined as:

\[
b^k \sum_{i=1}^{n} \frac{1}{g_i(X)} \quad \text{for} \quad g_i(X) > 0
\]

(D.12)
This is Fiacco-McCormick type of penalty function.

The augmented objective function for $g_i(X) > 0$ is:

$$P(X, b^k) = f(X) + b^k \sum_{i=1}^{n} \frac{1}{g_i(X)}$$

(D.14)

The penalty function can also be defined as:

$$- b^k \sum_{i=1}^{n} \ln g_i(X) \text{ for } g_i(X) > 0$$

(D.15)

$$b^k \sum_{i=1}^{n} \ln g_i(X) \text{ for } g_i(X) < 0$$

(D.16)

This is Lootsma type of penalty function.

The augmented objective function for $g(X) > 0$ is:

$$P(X, b^k) = f(X) - b^k \sum_{i=1}^{n} \ln g_i(X)$$

(D.17)

For the first method, a set of diminishing values of $b^k$ are selected, and $P(X, b^k)$ is minimized for each of them. A necessary condition before commencing the first minimization is that an initial feasible point $[X]^0$ has to be available, i.e., a point where all inequality constraints $g_i(X) > 0$ are satisfied. During the above process, the penalty term functions such that it stops $[X]$ from exceeding its limits. The description of the first method is equally
valid for the Lootsma penalty function method.

In the "Exterior SUMT method", the penalty term is defined as:

$$b^k \sum_{i=1}^{n} [g_i(X)]^2 \quad g_i(X) \geq 0$$

(D.18)

This is the Zangwill type of penalty function. With this type of penalty function, a sequence of $b^k$ are chosen from small values to infinitely large ones and $P(X,b^k)$ is minimized for each of them.

Of the three methods, the Zangwill method has the advantage over the other two in that it can handle both equality and inequality constraints and does not require an initial feasible point. The method works from both sides of the constraint region while the other two work only from the inside of the feasible region and depend on "a boundary repulsion effect" to keep the values of the vector $[X]$ from going out until minimal point is found. With the Zangwill method, the equality constraints and the violated inequality constraints are used to penalize the function being minimized. As soon as some inequality constraint is satisfied, it is removed from (D.18) until it is again violated. In the Optimal Power Flow problem, the inequality constraints are usually in the form of:

$$[X]_{\text{min}} \leq [X] \leq [X]_{\text{max}}$$

(D.19)

From the above analysis, it is seen that the Zangwill method is usually preferred.