A STUDY OF OPTIMAL LOAD FLOW USING NONLINEAR PROGRAMMING

by

Craig Alan Keeler

Thesis submitted to the graduate faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

in

Electrical Engineering

APPROVED:

L. L. Grigsby, Chairman

T. E. Bechert

R. H. Miller

May 1977

Blacksburg, Virginia 24061

ACKNOWLEDGEMENTS

The author would like to express his sincere thanks to Dr. L. L. Grigsby for serving as chairman of his graduate committee. Without his support, guidance and encouragement this work would not have been accomplished. Thanks are extended as well to Dr. T. E. Bechert and Dr. R. H. Miller who also served on the graduate committee.

The author is indebted to the Energy Research Group for their financial support which made his graduate study at Virginia Polytechnic Institute and State University possible. Thanks are due as well to the organizations who contribute to this very worthwhile program.

The author would like to express his very deep appreciation to who was very helpful and patient during the typing and correction of this thesis.

Finally, the author is grateful to his wife, , and his two girls, and , for sacrificing those evenings and week-ends spent in this endeavor.

TABLE OF CONTENTS

TABLE OF CONTENTS, cont.

LIST OF FIGURES

LIST OF TABLES

I. INTRODUCTION

1.1 Statement of the Problem

The optimal load flow is a static optimization problem. A scalar cost function which depends on power system quantities such as power generated and bus voltage is minimized. Restrictions, including upper and lower limits on these and other quantities, may be specified. The solution obtained is static since a new steady-state operating point is determined from some initial or assumed point with no dynamics considered.

There are many formulations to the optimal load flow problem. Among these are economic dispatch, minimum transmission loss, minimum reactive compensation and minimum pollution dispatch. The first two will be described in greater detail below. See reference [10] for a more complete bibliography of the possible problems. The solutions of the optimal load flow involve manipulations with complex, nonlinear equations, thus dictating the use of digital computers.

1.2 Scope of the Investigation

Several types of computer methods have been applied to solve the optimal load flow problem. Among these, linear, quadratic and nonlinear programming have received widespread attention. This research was limited to a detailed study of the nonlinear programming solutions to the optimal load flow problems. Three acclaimed techniques were then selected for computer implementation in order to determine which would be most practical.

Precautions were taken when programming to assure an unbiased comparison. All programs were prepared by the author in FORTRAN and executed at the Virginia Tech Computing Facility on the I.B.M. 370/158 operating system. Also, common routines were used for different programs whenever appropriate. For example, a load flow was required for some portion of all programs, and a common Newton-Raphson polar form routine was used for each one. To test the programs the IEEE standard 14, 30 and 57 bus systems [12] were used with cost function data taken from reference [11].

II. MATHEMATICAL FORMULATION OF THE OPTIMAL LOAD FLOW PROBLEM

2.1 The Load Flow Problem

The load flow may be viewed as a base for the optimal load flow since the initial and final points must satisfy the network power flow equations. The load flow is derived from these equations in general. As mentioned above, the Newton-Raphson polar form load flow is used for all techniques studied, and it is developed in some detail here [5]. See reference [14] or [15] for descriptions of other load flow methods.

The admittance form of the network power flow equations may be written in exponential form for an n-bus system,

$$P_k - jQ_k = \sum_{i=1}^{n} V_k Y_{ki} V_i \exp[-j(\delta_k + \psi_{ki} - \delta_i)] \qquad (2.1)$$

$$k=1,2,\ldots n$$

where,

P_k is the real power injected at bus k,

Q_k is the reactive power injected at bus k,

V_k is the voltage magnitude at bus k,

δ_k is the voltage angle at bus k,

Y_{ki} is the magnitude of the admittance connecting busses i and k,

$-\psi_{ki}$ is the angle of the admittance connecting busses i and k.

It should be noted that this form depends on the following definition of the bus admittances,

$$Y_{ki} \exp[-j\psi_{ki}] \triangleq G_{ki} - jB_{ki} \qquad (2.2)$$

$$k = 1,2,\ldots,n$$
$$i = 1,2,\ldots,n$$

where,

G_{ki} is the real part of the admittance connecting busses i and k, and B_{ki} is the imaginary part of the admittance connecting busses i and k. The equations (2.1) may be separated into real and imaginary parts as follows:

$$P_k = P_k^G - P_k^D = \sum_{i=1}^{n} V_k Y_{ki} V_i \cos(\delta_k + \psi_{ki} - \delta_i) \qquad (2.3)$$
$$k = 1, 2, \ldots, n$$

$$Q_k = Q_k^G - Q_k^D = \sum_{i=1}^{n} V_k Y_{ki} V_i \sin(\delta_k + \psi_{ki} - \delta_i) \qquad (2.4)$$
$$k=1, 2, \ldots, n$$

where P_k^G is the real power generated at bus k,

P_k^D is the real power demanded at bus k,

Q_k^G is the reactive power generated at bus k, and

Q_k^D is the reactive power demanded at bus k.

By expanding each equation (2.3) and (2.4) in a Taylor's series and retaining only the linear terms, the Newton-Raphson algorithm is obtained (see Appendix A):

$$\begin{bmatrix} \Delta P \\ \hline \Delta Q \end{bmatrix} = \begin{bmatrix} P_{sched} - P \\ \hline Q_{sched} - P \end{bmatrix} = \left[\begin{array}{c|c} \frac{\partial P}{\partial \delta} & V\frac{\partial P}{\partial V} \\ \hline \frac{\partial Q}{\partial \delta} & V\frac{\partial Q}{\partial V} \end{array}\right] \cdot \begin{bmatrix} \Delta\delta \\ \hline \frac{\Delta V}{V} \end{bmatrix} = \left[\begin{array}{c|c} H & N \\ \hline J & L \end{array}\right] \cdot \begin{bmatrix} \Delta\delta \\ \hline \frac{\Delta V}{V} \end{bmatrix} \qquad (2.5)$$

where ΔP is the vector of real power mismatches,

ΔQ is the vector of reactive power mismatches,

P_{sched} is the vector of scheduled real powers injected,

Q_{sched} is the vector of scheduled reactive powers injected,

$\Delta\delta$ is the vector of voltage angle corrections,

$\Delta V/V$ is the vector of voltage magnitude corrections, and the matrix of partial derivatives is referred to as the Jacobian matrix.

The following steps comprise the Newton-Raphson iteration:

i) All unknown voltage magnitudes and angles are initialized.

ii) The Jacobian matrix and power mismatch vectors are computed.

iii) The voltage correction vectors are found using some type of elimination method to avoid taking the inverse of the Jacobian.

iv) Update voltages and check for convergence against some error tolerance specified. If convergence is not obtained, return to step (ii) and continue.

v) Compute slack bus power and desired line flow powers.

There are variations to this procedure, especially in the handling of voltage-controlled busses, and some of these items will be taken up later. In section 2.3 the method for checking reactive power generation at voltage-controlled busses will be discussed.

2.2 The Static Optimization Problem

The static optimization problem can generally be stated as the solution which minimizes (or maximizes) a scalar cost function subject to equality and inequality constraints on the variables [16],

$$\underset{\text{w.r.t. } u}{\text{minimize}} \quad f(x,u) \tag{2.6}$$

$$\text{subject to} \quad h(x,u) = 0 \tag{2.7}$$

$$\text{and} \quad g(x,u) \leq 0 \tag{2.8}$$

where,

$f(x,u)$ is the scalar cost function,

$h(x,u)$ is the vector of equality constraints,

$g(x,u)$ is the vector of inequality constraints,

x is the vector of load flow variables, u is the vector of minimization variables,

and there are a fewer number of unknown variables, u, than equality constraints, h(x,u). The inequality constraints may consist of upper and lower bounds on the variables u and x. In addition, they may include functional constraints, depending on both u and x, for example, reactive power generation limits at the voltage-controlled busses.

One solution to the above problem depends on the transformation to an unconstrained optimization. The Lagrangian method can be used to accomplish this transformation. A new cost function, termed the Lagrangian, is formed,

$$L(x,u,\lambda,\mu) = f(x,u) + \lambda^T h(x,u) + \mu^T g(x,\mu) \qquad (2.9)$$

where,

$L(x,u,\lambda,\mu)$ is the Lagrangian function,

λ is the vector of Lagrange multipliers, and

μ is the vector of dual variables.

Special caution must be taken when handling inequality constraints since they are included in the Lagrangian function only when they are violated, $g_i(x,u) > 0$. This fact is expressed in the exclusion equation,

$$\mu^T g(x,u) = 0; \ \mu \geq 0. \qquad (2.10)$$

The minimization of $L(x,u,\lambda,\mu)$ will result in the minimum point of f(x,u) and will satisfy the constraints. Other means of handling inequality constraints, such as the penalty functions method, are described in conjunction with specific programming techniques.

Some necessary conditions for the minimum point are given by the Kuhn-Tucker theorem[16]:

$$\frac{\partial L}{\partial x} = 0 = \frac{\partial f}{\partial x} + \lambda^T \frac{\partial h}{\partial x} + \mu^T \frac{\partial g}{\partial x} \tag{2.11a}$$

$$\frac{\partial L}{\partial u} = 0 = \frac{\partial f}{\partial u} + \lambda^T \frac{\partial h}{\partial u} + \mu^T \frac{\partial g}{\partial u} \tag{2.11b}$$

$$\frac{\partial L}{\partial \lambda} = 0 = h(x,u) \tag{2.12}$$

$$0 = \mu^T g(x,u), \quad \mu > 0 \tag{2.13}$$

Note the similarity between equations (2.7) and (2.8) with (2.12) and (2.13). Equation (2.11) turns out to be most useful in solving for the Lagrange multipliers, λ. These conditions have been applied to various techniques for solving the optimal load flow problem [9], described in the next section.

2.3 The Optimal Load Flow Problem

The development of the general optimization problem above can be applied to the optimal load flow problems of economic dispatch and minimum transmission loss [1]. Throughout this discussion the load flow variables are

$$x = \begin{bmatrix} P \\ Q \\ V \\ \delta \\ Q \\ \delta \end{bmatrix} \begin{matrix} \text{, at the slack bus} \\ \\ \text{, at the load busses} \\ \\ \text{, at the voltage-controlled busses.} \\ \\ \end{matrix} \tag{2.14}$$

The remaining system quantities are lumped under the minimization variables,

$$u = \begin{bmatrix} V \\ \delta \\ P \\ Q \\ P \\ V \end{bmatrix} \begin{matrix} \text{, at the slack bus} \\ \\ \text{, at the load busses} \\ \\ \text{, at the voltage-controlled busses.} \\ \\ \end{matrix} \tag{2.15}$$

Certain of the u variables are selected for adjustment, depending on the specific problem at hand.

For the economic dispatch problem the minimization variables are all adjustable real power sources, P_i^G. The cost function used for dispatching p real power sources is

$$f(x,u) = \sum_{i=1}^{p} f_i(P_i^G) \tag{2.16}$$

where,

$f_i(P_i^G)$ is the quadratic cost function for bus i.

The adjustable real power sources may be located at any load or voltage-controlled bus, as well as at the slack bus. Note that a cost must be assigned to the slack bus so that the minimization will not try to assign all generation to it. The equality constraints for the economic dispatch problem are one equation (2.3) for each bus with fixed real power injected and one equation (2.4) for each load bus. Inequality limit constraints are normally placed on the p adjustable real powers,

$$P_i^\ell \leq P_i^G \leq P_i^u \qquad i=1,2,\ldots,p \tag{2.17}$$

where,

P_i^u is the upper limit of real power generation for bus i,

P_i^ℓ is the lower limit of real power generation for bus i.

Limits are usually specified for reactive power generation, Q_i^G, at the m voltage-controlled busses also,

$$Q_i^\ell \leq Q_i^G \leq Q_i^u \qquad i=1,2,\ldots,m \tag{2.18}$$

where,

Q_i^u is the upper limit of reactive power generation for bus i,

Q_i^ℓ is the lower limit of reactive power generation for bus i.

These limits on reactive power generation may be handled in one of two ways. In the load flow the reactive power is clamped at the limiting value whenever a violation would occur. This method can be carried over to the optimal load flow, called a "hard" constraint. The voltage-controlled bus at which this occurs must then be type changed to a load bus as in the typical load flow handling. The alternate method introduces functional inequality constraints, equation (2.4) for each voltage-controlled bus with Q_i^G limits violated. In addition, the penalty functions method, discussed later, may be used, called a "soft" constraint.

Finally, limits may be specified on voltage magnitude at any or all of the voltage controlled or load busses,

$$V_i^\ell \leq V_i \leq V_i^u \qquad i=1,2,\ldots,n \tag{2.19}$$

where,

V_i^u is the upper limit of voltage magnitude for bus i,

V_i^ℓ is the lower limit of voltage magnitude for bus i.

These limits are useful in maintaining a uniform voltage profile over the entire power system but may be difficult to satisfy. Typically, these constraints are handled by the soft penalty function method.

For the minimum transmission loss or reactive power dispatch the minimization variables are all controllable bus voltage magnitudes. The cost function used for dispatching m voltage-controlled busses and the slack bus is

$$f(x,u) = P_s(V,\delta) \tag{2.20}$$

where,

$P_s(V,\delta)$ is equation (2.3) for the slack bus.

This function associates transmission loss with the reactive power dispatch since in general it is difficult to assign a cost to production of reactive power unless, say, reactive compensation was to be installed at load busses for some cost. The equality constraints for the minimum loss problem are one equation (2.3) for each bus except the slack bus and one equation (2.4) for each load bus. Inequality limit constraints are normally placed on the m+1 controllable voltages, equation (2.19). Reactive power generation limits are usually specified at the m voltage-controlled busses, equation (2.18). Limits may also be specified on load bus voltage magnitudes, equation (2.19), as in the economic dispatch.

2.4 Summary

The basis for the optimal load flow has been shown to be the conventional load flow. The problems of real and reactive power dispatch have been developed from the static constrained optimization problem. The transformation to unconstrained minimization has been discussed along with some necessary conditions for solution. The optimal problems have been posed with cost functions and constraint equations. Solution techniques will be described in the next chapter.

III. NONLINEAR PROGRAMMING SOLUTIONS TO THE OPTIMAL LOAD FLOW PROBLEM

3.1 Review of Early Solutions

In the late 1960's many utilities in the country had digital computers on-line to economically schedule real power generation using the B constants method.[13] This method minimized the cost function equation (2.16) for p real power sources, subject to the real power balance equation for an n-bus system

$$\sum_{i=1}^{n} P_i^D + P^L - \sum_{i=1}^{p} P_i^G = 0 \qquad (3.1)$$

where P^L is the real power losses in the system.
The B constants arise in the calculation of P^L from a loss formula such as

$$P^L = B_o + (P^G)^T B_1 + (P^G)^T B_2 P^G \qquad (3.2)$$

where P^G is a vector of real power generations,

B_o is a scalar,

B_1 is a vector, and

B_2 is a matrix.

The B_1 values are constant for a fixed system configuration and, thus, must be recalculated for any change in the system. The method is restricted to the adjustment of real power generations subject to limits as given in equation (2.17). Applying the Kuhn-Tucker condition equation (2.11) to the Lagrangian function equation (2.9) for p real power sources results in

$$\frac{\partial L}{\partial P_i^G} = 0 = \frac{\partial f_i}{\partial P_i^G} + \lambda\left(\frac{\partial P^L}{\partial P_i^G} - 1\right), \quad i=1,2,\ldots,p \qquad (3.3)$$

where $\partial f_i / \partial P_i^G$ is the incremental cost and

$(\frac{\partial P^L}{\partial P_i^G} - 1)$ is the penalty factor.

This method has achieved widespread application with considerable success, but it lacks the flexibility to solve the more general formulation of the optimal load flow as developed in section 2.3 above.

A similar method was later developed by Dopazo, Klitin, Stagg and Watson [6] which replaced the approximate loss formula equation (3.2) with an exact equation derived from

$$P^L + jQ^L = (I^*)^T Z I \tag{3.4}$$

where Q^L is the reactive power losses in the system,

I is the vector of bus currents, and

Z is the bus impedance matrix.

The exact loss equation then becomes

$$P_L = \begin{bmatrix} P \\ \hline Q \end{bmatrix}^T \left[\begin{array}{c|c} \alpha & \beta \\ \hline -\beta & \alpha \end{array}\right] \begin{bmatrix} P \\ \hline Q \end{bmatrix} \tag{3.5}$$

where P is the vector of net real bus powers and

Q is the vector of net reactive bus powers.

The α and β elements are calculated from the equations

$$\alpha_{ij} = \frac{r_{ij}}{V_i V_j} \cos(\delta_i - \delta_j) \tag{3.6}$$

$$\beta_{ij} = - \frac{r_{ij}}{V_i V_j} \sin(\delta_i - \delta_j) \tag{3.7}$$

where r_{ij} is the resistance component of the bus impedance matrix ij^{th}

element. When the Kuhn-Tucker conditions are applied to this formulation, the result is again equation (3.3); however, the terms $(\partial P^L/\partial P_i^G)$ are now exact expressions and referred to as the incremental transmission losses. An extension schedule reactive power generation

$$\frac{\partial L}{\partial Q_i^G} = 0 = \frac{\partial f_i}{\partial Q_i^G} + \lambda \frac{\partial P^L}{\partial Q_i^G} \tag{3.8}$$

however $f_i(P_i^G)$ is now a function of Q_i^G so that this equation reduces to

$$\frac{\partial L}{\partial Q_i^G} = 0 = \lambda \frac{\partial P^L}{\partial Q_i^G} \tag{3.9}$$

While this method extended can adjust reactive power generations subject to limits as given in equation (2.18), the bus voltage magnitudes are not scheduled, hence a minimum loss dispatch cannot be assured [9]. This method does provide a prelude to the optimal load flow solution.

3.2 Steepest Descent Minimization

The method of steepest descent is probably the most popular and widely used technique for optimizing nonlinear problems of several variables. Ignoring constraints, the method is to compute iteratively a succession of control parameters from the equation

$$u^{(i+1)} = u^{(i)} - c^{(i)} \nabla f(x, u^{(i)}) \tag{3.10}$$

where $u^{(i+1)}$ is the succeeding vector of control parameters,

$u^{(i)}$ is the last vector of control parameters,

$\nabla f(x,u^{(i)})$ is the present gradient of the cost function, $\partial f(x,u)/\partial u$,

and $c^{(i)}$ is a nonnegative scalar which must be selected according to the equation

$$\underset{\text{w.r.t. } c^{(i)}}{\text{minimize}} f(x, u^{(i+1)}) \tag{3.11}$$

That is, u is adjusted according to a line search along the negative gradient, $-\nabla f(x,u)$, and c is the distance to the minimum in that direction (see Figure 1). However, the beautiful simplicity of this method is obscured when constraints are considered, since f and g become complicated equations with many varied terms.

This method was first applied to solve the optimal load flow by Dommel and Tinney [1], who formulated the problem in the general form of section 2.3. This method provides a much more flexible solution than had previously been possible. A choice of cost functions is now available. For example, to minimize cost and losses simultaneously the cost equations is

$$f(x,u) = \sum_{i=1}^{p} f_i(P_i^G) + P_s(V,\delta) \tag{3.12}$$

subject to the network power flow equations. Control parameters for this problem are the p adjustable real power sources for which cost equations, f_i, have been described and m controlled voltages. Hard limit constraints are normally placed on these quantities except for the following variable. The slack power is dependent on the network power flow equations and thus cannot be constrained beforehand due to unknown system losses. When this control variable exceeds, say, its upper limit, a penalty function can be added to the cost equation of the form

$$w_s(P^G) = r_s(P_s^G - P_s^u)^2 \tag{3.13}$$

where r_s is a nonnegative, scalar penalty factor for the slack bus power.

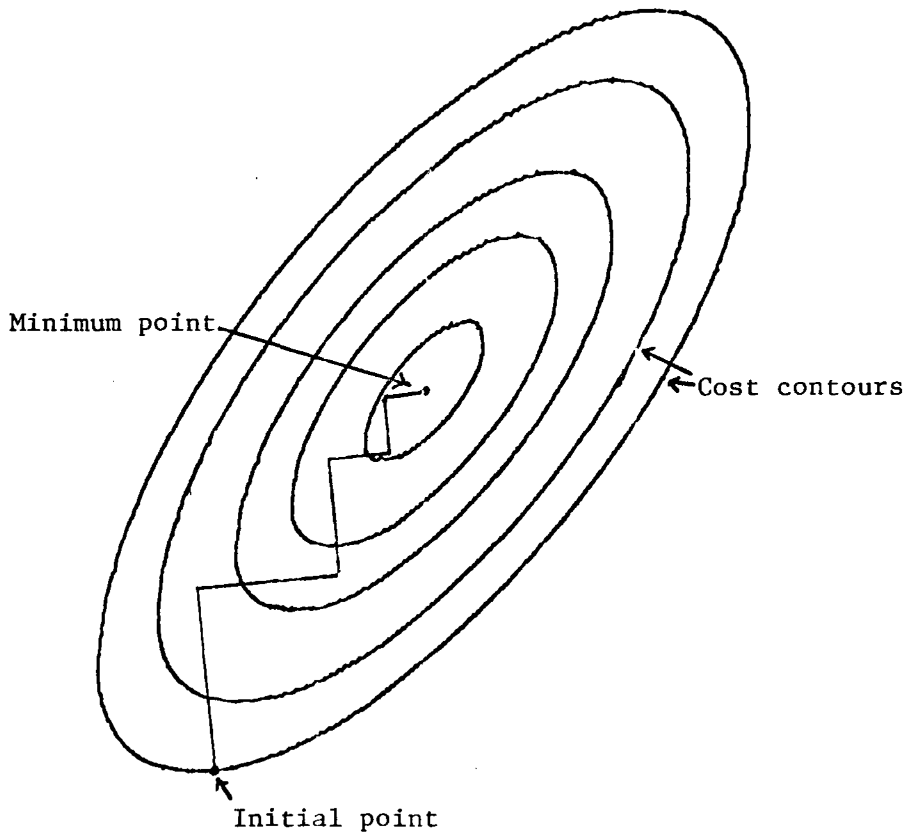

Figure 1. Steepest Descent Minimization

This equation provides a soft limit or constraint which brings $P_s^G(V,\delta)$ back near its upper limit (see Figure 2). The relative magnitude of r_s determines how much the slack bus power is allowed to deviate from its hard limit. To decrease this deviation, the value of the penalty factor should be increased.

Similar penalty functions are used to constrain reactive power generations which have been violated at voltage controlled busses. A violation of the lower limit at bus i, say, would result in a penalty function of the form

$$w_i(Q_i^G) = s_i(Q_i^G - Q_i^\ell)^2 \tag{3.14}$$

where s_i is penalty factor.

Finally, penalty functions are also used when it is desired to constrain the voltage magnitude within limits at any or all of the load busses. A penalty function for a violation of the upper limit at bus j, say, would have the form

$$w_j(V_j) = t_j(V_j - V_j^u)^2 \tag{3.15}$$

where t_j is the penalty factor.

It should be noted that the penalty function for a soft limit should be added to the cost function, f(x,u), only if it is violated or "active". All inactive penalty functions should be omitted from the cost function before the minimization procedure begins.

The scalar cost function, or Lagrangian, to minimize costs and losses thus formed with the addition of penalty functions becomes

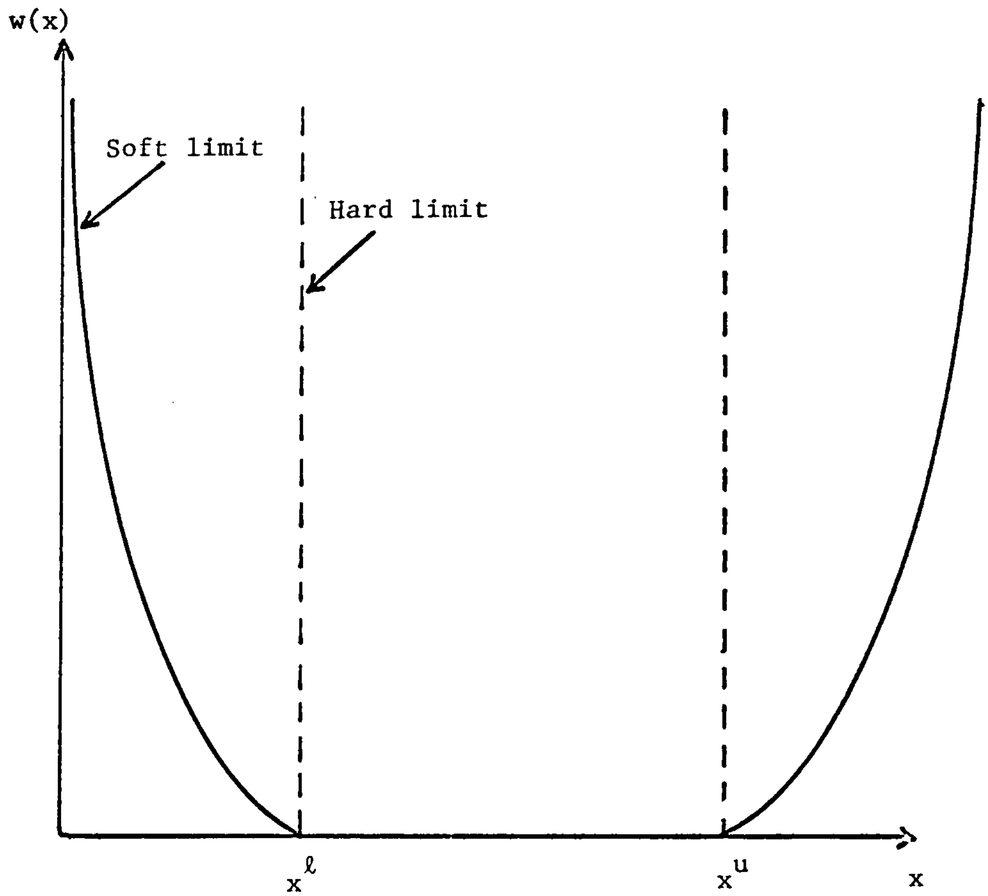

Figure 2. Penalty Function for Soft Limit Constraints

$$L(x,u) = f(x,u) + [\lambda^P \ \lambda^Q]^T \begin{bmatrix} P(V,\delta) \\ \text{------} \\ Q(V,\delta) \end{bmatrix} +$$

$$[\mu^P \ \mu^Q]^T \begin{bmatrix} (P^G-P^\ell) \text{ or } (P^G-P^u) \\ (V-V^\ell) \text{ or } (V-V^u) \end{bmatrix} + \sum_i w_i(P_s^G, Q^G, V) \tag{3.16}$$

The gradient, or vector of first partial derivatives, of this equation must be taken with respect to all variables to satisfy the Kuhn-Tucker necessary conditions for the minimum. Details of this derivation for the minimum costs and losses case are found in Appendix B. Since hard constraints are maintained on all control variables, the dual variables, μ's do not need to be evaluated in the minimization procedure and are simply set equal to zero.

The following steps comprise the steepest descent optimal load flow:

i) Initialize all control voltages and real power generations.

ii) Perform a load flow by the Newton-Raphson technique and save the final Jacobian matrix for use in step (iv).

iii) Determine all active inequality constraints at the solution point and augment the cost function with penalties.

iv) Solve for the Lagrange multiplier, using the gradient w.r.t. the load flow variables, equation (2.11b).

v) Solve for the control variable corrections using the gradient w.r.t. the control variables, equation (2.11a), adjust control variables using equation (3.10), and check for convergence against some error tolerance. If convergence is not obtained, return to step (ii) and continue.

vi) Compute final costs and losses at the minimum.

The critical step which determines the success of the minimization procedure is step (v) where the control variables are adjusted. Several possible methods have been suggested to accomplish this, including a line search, or one-dimensional minimization, which was selected. In this step the cost function is approximated by the quadratic costs, equation (2.16), to become

$$\underset{\text{w.r.t.c}}{\text{minimize}}\ \Sigma_i f_i(P_i^G - c\nabla f_i^P) \tag{3.17}$$

where ∇f_i^P are the gradient terms for the real power generations and c is the distance moved in the feasible direction (see Figure 3). A final modification is possible when control variable corrections change sign in succeeding iterations, indicating their proximity to the minimum point. It has been found that by using a backward difference formula to approximate the second partial derivatives of the cost function, or Hessian terms, improved control variable corrections can be obtained. This can be calculated for the k^{th} iteration from

$$H_i^{(k)} = \frac{\nabla f_i^{(k-1)} - \nabla f_i^{(k)}}{u_i^{(k-1)} - u_i^{(k)}} \tag{3.18}$$

where $H_i^{(k)}$ is the Hessian term at the k^{th} iteration.
The control variable corrections then become, for $H_i^{(k)}$ positive,

$$\Delta u_i^{(k)} = -\frac{\nabla f_i^{(k)}}{H_i^{(k)}} \tag{3.19}$$

otherwise $\Delta u_i^{(k)}$ should be set to zero since a descent step will not be assured.

The performance of this method will be evaluated in the next chapter when results for actual systems taken from reference [12] are presented.

3.3 Fletcher-Powell Minimization

The Fletcher-Powell method[16] was among the earliest to form the inverse Hessian matrix of second partial derivatives for a nonlinear function f(y). An iteration scheme is used which requires the evaluation of the gradient vector of first partial derivatives of the function at each step in the minimization. Successive corrections are then made to the variables by Newton's method:

$$\Delta y = -(\nabla^2 f(y))^{-1}(\nabla f(y)) \tag{3.20}$$

where Δy is the variable correction vector,

$\nabla^2 f(y)$ is the Hessian matrix, and

$\nabla f(y)$ is the gradient vector.

If the function f(y) were a quadratic, this procedure would arrive at the minimum point in one step from any starting point. However, for most problems a series of adjustments must be made and the inverse Hessian and gradient calculated at each step.

The procedure to form the inverse Hessian by the Fletcher-Powell method is as follows:

i) Initialize the inverse Hessian to any symmetric positive definite matrix, for example, the identity matrix and calculate the gradient vector.

ii) Calculate the correction vector, or feasible direction, from equation (3.20) above.

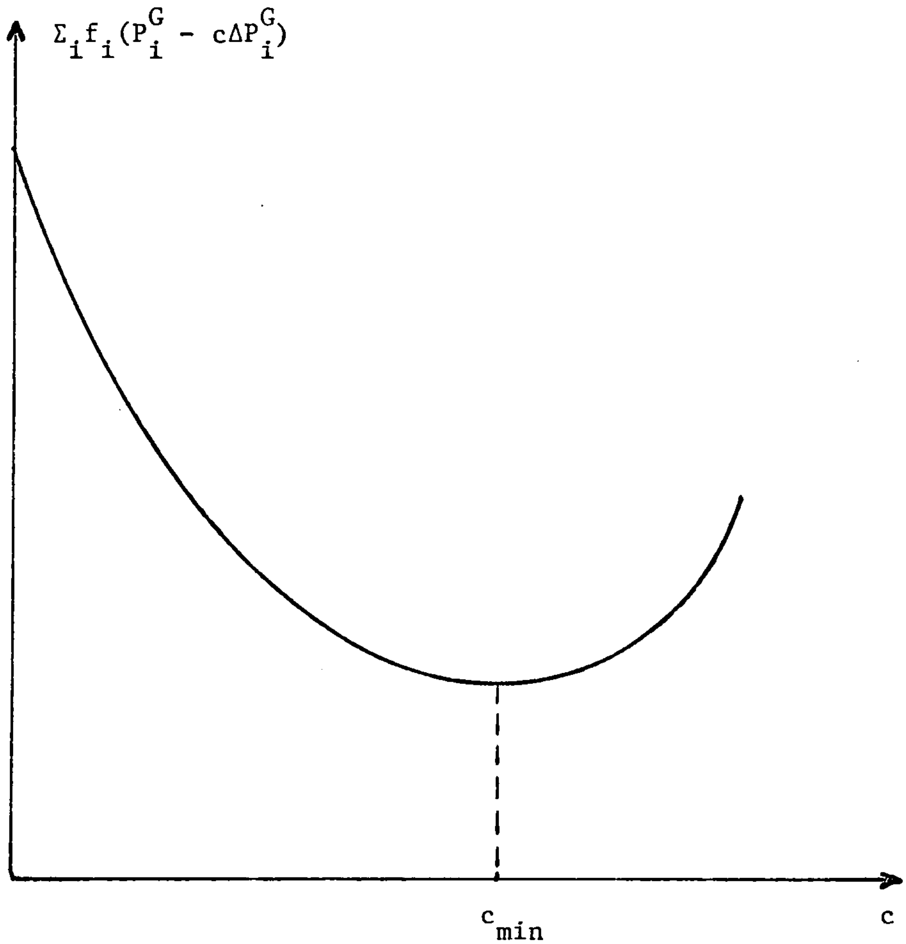

Figure 3. Quadratic Fit Line Search

iii) Perform a line search to optimally adjust the y variables, equation (3.17), and compute the new gradient vector.

iv) Compute the inverse Hessian correction matrix according to the Fletcher-Powell algorithm[17] and check terms for convergence against some prescribed error tolerance. If it is not obtained, return to step (ii) with the new gradient vector and continue.

v) Evaluate the cost function, f(y), at the minimum point.

This method was first applied to the optimal load flow solution by A. M. Sasson[2]. In this early work several techniques were considered for transformation from constrained to unconstrained minimization. Methods due to Fiacco-McCormick, Lootsma, and Zangwill were discussed, but none were found to have special advantage for this problem. In his later work[3], Sasson develops the method of Powell to complete this solution of the optimal load flow. Powell's method, similar to the method of penalty factors, augments the cost function with terms related to the equality and active inequality constraints

$$L(y,s,r) = f(y) + \Sigma_i \frac{(h_i(y)+s_i)^2}{r_i} + \Sigma_j \frac{(g_j(y)+s_j)^2}{r_j} \qquad (3.21)$$

where s_i and r_i are nonnegative scalars associated with the equality constraints $h_i(y)$ and

s_j and r_j are nonnegative scalars associated with the inequality constraints $g_j(y)$.

By successive adjustment of the s and r factors at each step in the minimization, the larger constraint violations can be more severely penalized. Thus the convergence of the method can be improved. This method is more powerful than the method of steepest descent described in the previous section. The square error type terms in equation (3.21) involving the equality constraints, h(y), actually compell each successive point in the minimization to lie within the load flow solution space. Thus the requirement for intermediate load flow solutions during the minimization procedure is eliminated. For this reason Fletcher-Powell has been referred to as the first completely nonlinear optimal load flow solution technique. Further, by setting the cost equation, f(y), to zero, this method will solve the conventional load flow problem.

The minimization variables, y, which are adjusted by the Fletcher-Powell optimal load flow are the complex bus voltages only. All cost and constraint equations dependent on real and reactive powers are written in terms of these variables using equation (2.1) in rectangular form:

$$P_k = \sum_{i=1}^{n} [E_k(E_iG_{ki} + F_iB_{ki}) + F_k(F_iG_{ki} - E_iB_{ki})] \qquad (3.22)$$
$$k = 1, 2, \ldots, n$$

$$Q_k = \sum_{i=1}^{n} [F_k(E_iG_{ki} + (F_iB_{ki}) - E_k(F_iG_{ki} - E_iB_{ki})] \qquad (3.23)$$
$$k = 1, 2, \ldots, n$$

where E_i is the real part of the complex voltage at bus i and F_i is the imaginary part.

The constraint equations for bus voltage magnitude are written in terms of E_i and F_i by using the equation

$$V_i = \sqrt{E_i^2 + F_i^2}. \qquad (3.24)$$

The gradient vector of equation (3.21) is taken with respect to all E_i and F_i, except F_s at the slack bus, since the reference angle is zero. Details of the gradient terms are found in Appendix C.

The procedure of the Fletcher-Powell optimal load flow is as follows:

i) Select an initial point and solve the load flow problem using any method available. Set the initial inverse Hessian equal to the identity matrix.

ii) Perform the Fletcher-Powell method to construct the inverse Hessian matrix by successively forming the gradient of equation (3.21) and updating the minimization variables.

iii) Check the minimization variables for convergence against some prescribed error tolerance. If it is not obtained, adjust the r_i and s_i factors by the Powell method and return to step (ii) to recalculate the inverse Hessian.

iv) Compute final costs at the minimum point.

It has been found that by using the previous value of the inverse Hessian from step (ii) when returning from step (iii) to recalculate the matrix, then convergence is considerably faster than if the initial identity matrix is used. The line search used in step (ii) is similar to the one employed in the method of steepest descent; however, in this method the control variables are not directly adjusted. Therefore, a cubic line fit is used to approximate the cost function. Once the feasible direction has been found from equation (3.20), successive steps of distance c are taken until the minimum point is enclosed. This is detected by computing the norm of the gradient at each step from the equation

$$|\nabla f(y)| = \Delta y^T \nabla f(y + c\Delta y) \tag{3.25}$$

where $|\nabla f(y)|$ is the scalar norm and

c is the total distance moved in the feasible direction Δy.

When the norm of the gradient changes sign in succeeding steps, then the minimum point is enclosed. The value of c_{min} can now be calculated from the values of the cost function and gradient at each of the neighboring points (see Figure 4)[16].

Details of the Powell method of adjusting the r_i and s_i factors will be addressed in the next chapter when the performance of this method is described.

3.4 Hessian Matrix Minimization

Experience with the Fletcher-Powell method of constructing the inverse Hessian matrix led Sasson, Viloria, and Aboytes to try direct evaluation of the Hessian matrix for a specific cost function[4]. Then equation (3.25) above can be rewritten as

$$(\nabla^2 f(y))\Delta y = -\nabla f(y) \tag{3.26}$$

and some form of matrix elimination used to evaluate Δy. However, the form of the cost function, equation (3.21), with Powell's method of transformation applied, presents an imposing problem to differentiate and evaluate. To overcome this, a simplified approach to handle constraints was taken. An augmented cost function was used which consisted of the cost equation f(y) plus weighted sums of square error type constraint terms

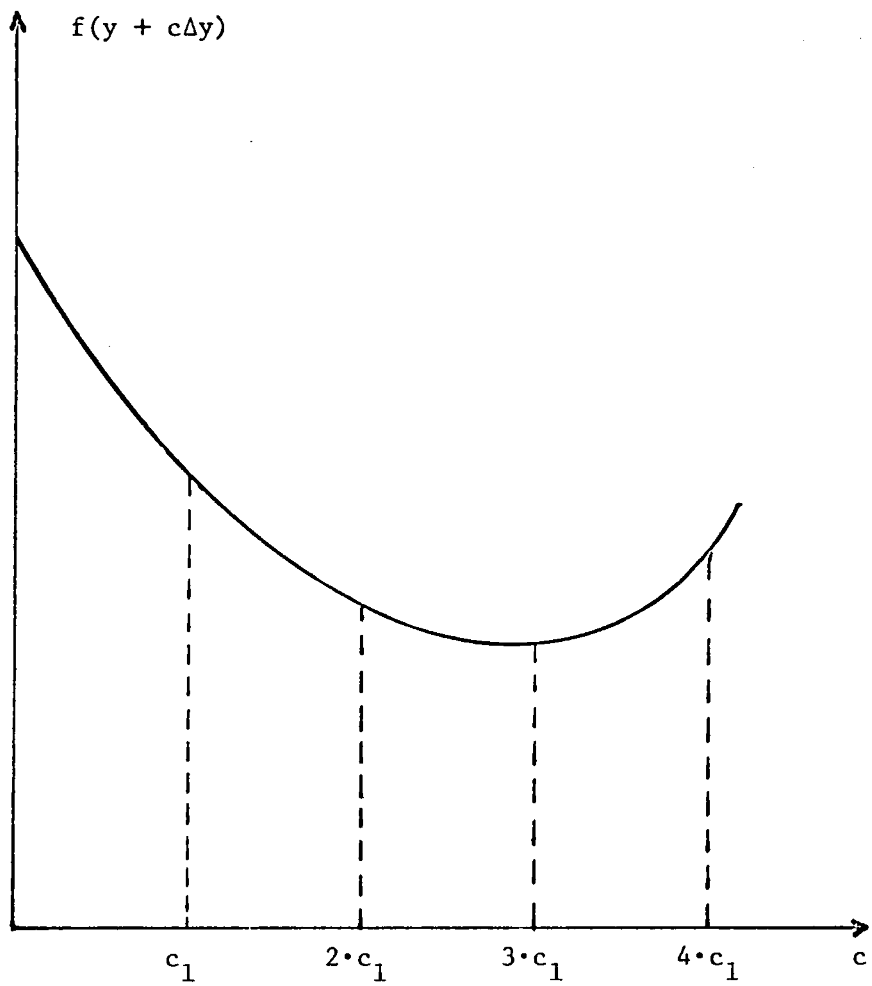

Figure 4. Cubic Fit Line Search

$$L(y,t) = f(y) + \Sigma_i t_i (h_i(y))^2 + \Sigma_j t_j (g_j(y))^2 \qquad (3.27)$$

where t_i are nonnegative scalar factors for equality constraints $h_i(y)$ and t_j are nonnegative scalar factors for inequality constraints $g_j(y)$. The Hessian matrix method closely parallels the Fletcher-Powell minimization in several respects. Here the minimization variables are, again, the complex bus voltages; however, in this case they are separated into the voltage magnitude and angle components. The Hessian matrix of second partial derivatives and the gradient vector of the cost function can be developed in terms of the Jacobian matrix of first partial derivatives in polar form.

The procedure for the Hessian matrix optimal load flow is as follows:

i) Select an initial point and obtain a load flow solution using the Newton-Raphson method in polar form. Store the final Jacobian matrix for use in step (ii).

ii) Evaluate the gradient and the Hessian from the terms of the Jacobian matrix.

iii) Calculate the feasible direction from equation (3.26) using some form of matrix elimination.

iv) Perform a line search to determine the optimal correction vector.

v) Check the convergence of the voltage correction against some prescribed error tolerance. If it is not obtained, adjust the t_i factors and return to step (ii).

vi) Evaluate the cost function at the minimum.

The line search used in step (iv) is the cubic fit scheme developed in the last section (see Figure 4). The t_i factors are simply increased at each iteration if the associated constraint is violated.

Further details of this procedure are included in the next chapter when the program results are discussed.

3.5 Summary

The nonlinear programming methods considered for the optimal load flow solution have been presented. Some of the mathematical details are developed further in the appendices. For a more complete explanation of any of the methods described here, refer to the original works.

IV. COMPUTER PROGRAMMING RESULTS AND COMPARISONS

4.1 Computer Programs

In the preparation of these computer programs, it was important that care be taken to assure, as much as possible, fair comparisons. A main program was written for each method to direct the execution of various subroutines. These subroutines then contain the various calculations which are performed; for example, construction of bus admittance matrix, Jacobian matrix, line searches, etc. The data is stored in common blocks which are accessed by each subroutine. This approach permits the direct substitution of those subroutines which are required by each method.

The data for the examples tested was the same for all methods. Initial starting values were also equal, except for the various penalty factors which were handled individually. Therefore, a common data initialization routine was written. Also, the bus admittance matrix was required for each method. Since no additional calculations, such as long line models, were needed, the bus admittance matrix was formed in the initialization routine. This eliminated the need to store the line data. The initial bus voltages are assigned values in this routine so that a load flow can be solved. A flat start at each bus of one per unit voltage magnitude and zero phase angle referred to the slack bus was used for all examples. Finally, the control parameters must be assigned values before the initial load flow can be solved. For the minimum costs and losses case, these are the adjustable real power generations and the controllable voltage magnitudes. These voltage magnitudes are left at one per unit unless their upper or lower limit is violated. If this occurs, they are set to their lower limit. The initial real power generations must be chosen so that demand is met without limit violations, except perhaps at the slack bus. This is done by hand calculation prior to the computer run and supplied to the initialization routine as input data.

The next step in each of the programs is the initial load flow. This is required to begin each minimization at a solution point of the system; however, some inequality constraints may be violated. Although the two completely nonlinear methods are capable of solving the load flow through a minimization technique with cost equations set to zero the Newton-Raphson load flow in polar form was used for all methods for the following reasons.

i) The Newton-Raphson load flow is required in polar form as a step in each iteration of the steepest descent minimization approach.

ii) To solve the load flow by the Fletcher-Powell minimization requires that the inverse Hessian matrix be formed from an initial value such as the identity matrix. This is a lengthy procedure which requires more storage when compared with the Newton-Raphson approach.

iii) The Hessian matrix minimization can be used to obtain a load flow solution; however, to form the Hessian matrix, terms from the Jacobian are calculated. Therefore, it is more convenient to use Newton-Raphson with fewer computations.

The steepest descent and Hessian matrix methods each requires the solution of linear systems of equations so a common triangular factorization routine and a back substitution routine were used. This was chosen since the steepest descent method required reuse of the factored Jacobian from the load flow in the solution of the Lagrange multipliers. The Fletcher-Powell and Hessian matrix methods each utilizes the cubic fit line search and a common routine was also used for this one-dimensional minimization.

Simplified flow charts of the computer programs are shown in Figures 5-8. In these figures each block represents the individual subroutines used. The main program for each method performs the indicated convergence tests and controls the sequence of operations. Although some of the subroutines not mentioned above appear to be required by more than one method, recall from the developments in Chapter Three that the augmented cost functions for each of the minimization methods differed. Thus the gradients will differ for each method, as can be seen in the appendices. This is also true for the penalty factor adjustments in each method. Penalty factors are adjusted to increase the corrections on minimization variables which still have large limit violations after one or two iterations. However, the penalty functions used to augment the cost functions in each method differ with regard to factors and variables.

The final step in each procedure is then to compute the final cost and print important data, such as the time required for execution which is of interest here.

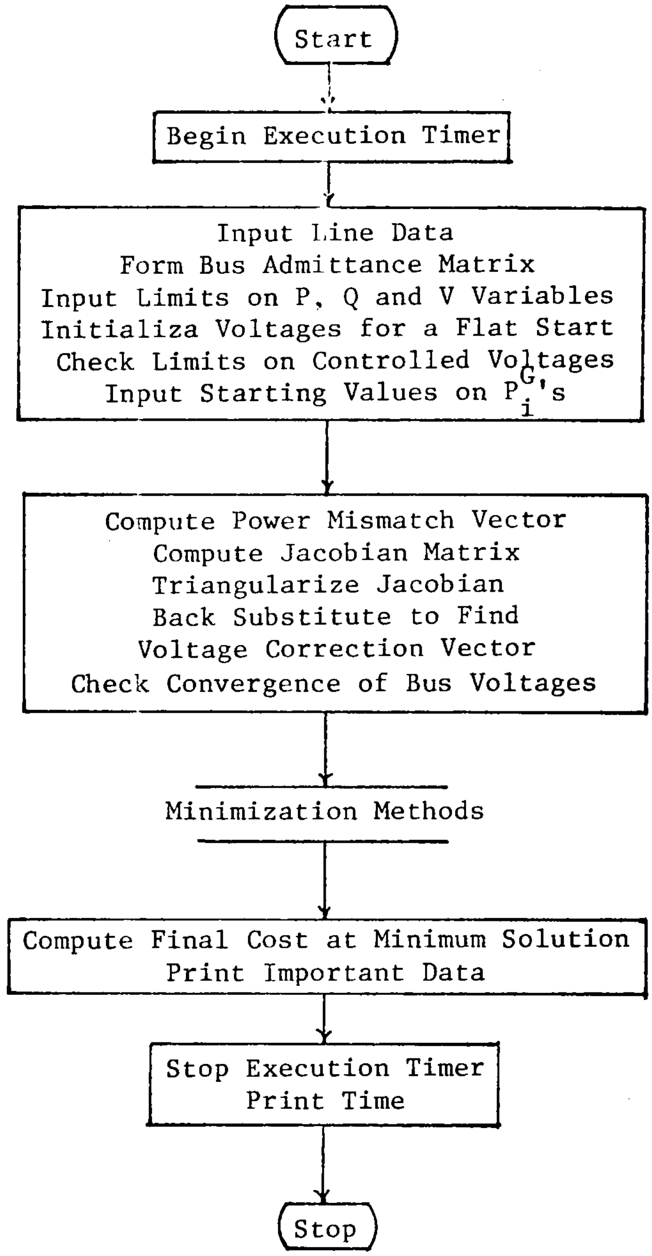

Figure 5. Flow Chart of Common Initialization, Load Flow and Final Routines

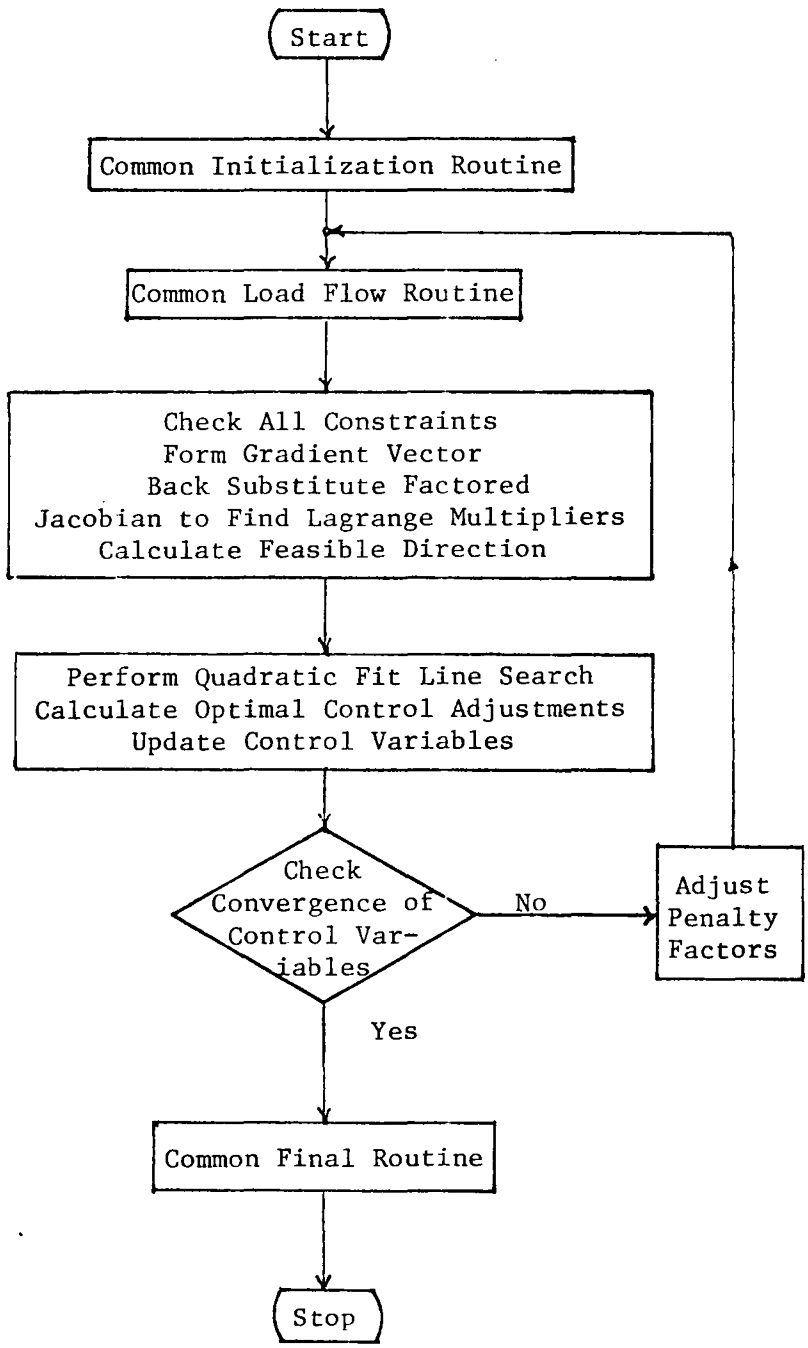

Figure 6. Flow Chart for Steepest Descent Optimal Load Flow

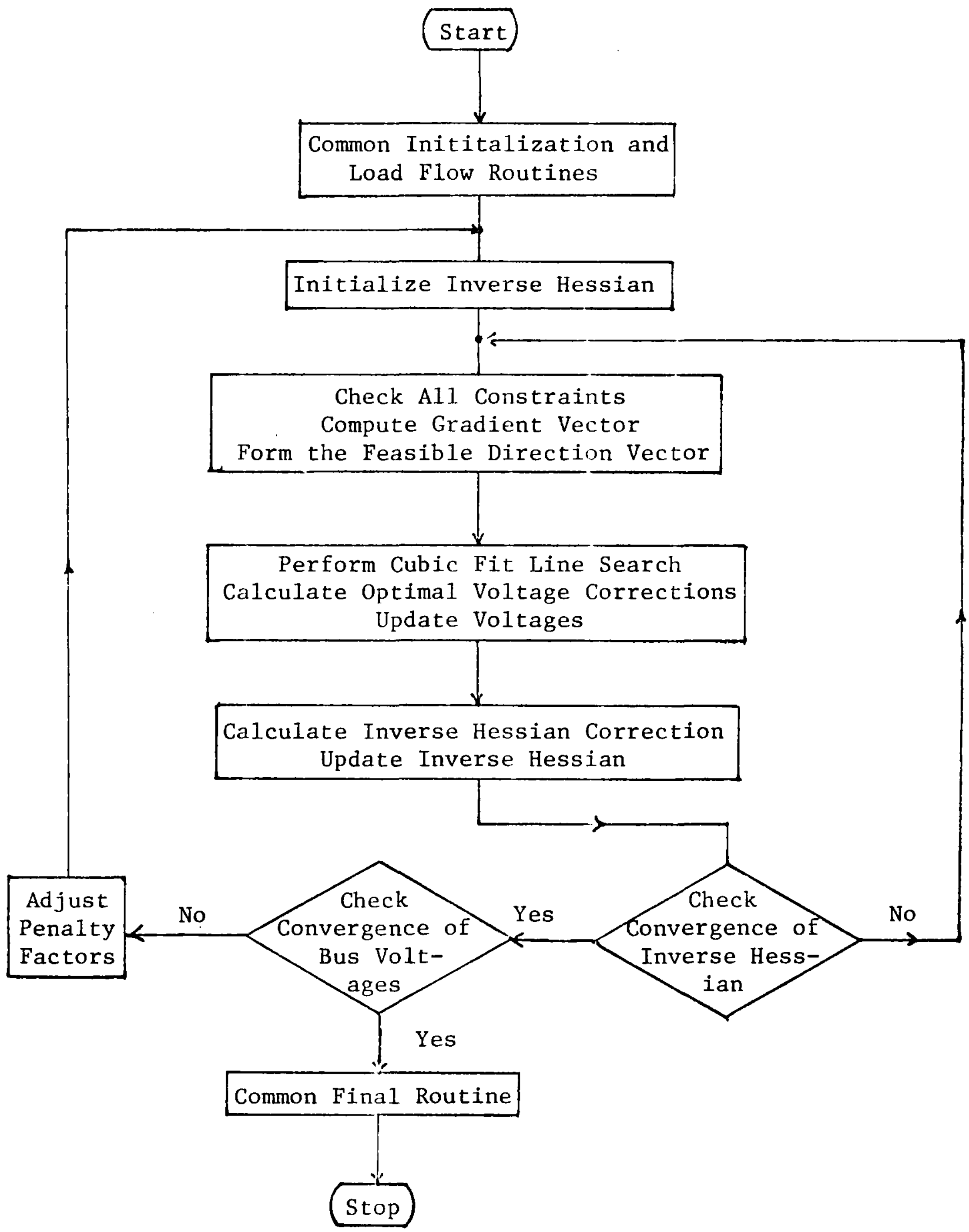

Figure 7. Flow Chart for Fletcher-Powell Optimal Load Flow

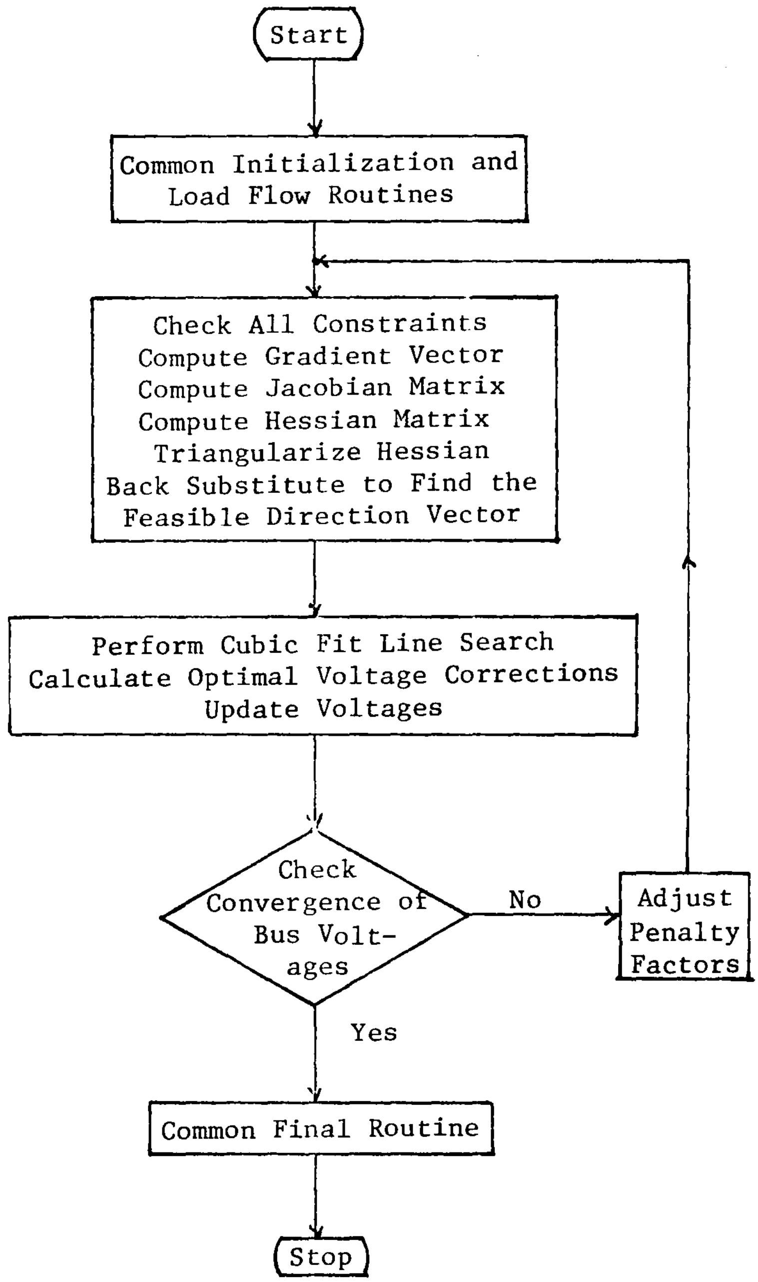

Figure 8. Flow Chart for Hessian Matrix Optimal Load Flow

4.2 Penalty Factor Adjustment

As seen in the flow charts of the previous section, all networks have the capability of adjusting penalty factors to more strictly penalize constraint violations. Each method employs a different mechanism to effect these adjustments. The relative magnitude of the penalty terms must be kept small in each cost function when compared with the cost equations. If these penalty terms become too large, convergence will be affected.

The following are techniques employed by the methods to adjust the penalty factors when limit violations are not reduced fast enough.

i) In the steepest descent optimal load flow all penalty factors are initially set equal to 10. If limits are not met on inequality constraints after two iterations, they are increased by a factor of 5 or 10, depending on the amount of the violation at each active constraint.

ii) The Fletcher-Powell optimal load flow requires the adjustment of two penalty factors for each active constraint. Initially, all r_i factors are set to $\frac{1}{20}$ and s_i to zero. If the constraint violations are small after each iteration, the s_i factors are set equal to the magnitude of the violations. If the violations are large, both the r_i and the s_i factors are reduced by a factor of 100.

iii) The Hessian matrix optimal load flow takes a similar approach as was taken with steepest descent. All penalty factors are initially set equal to one. Then after two iterations, factors associated with active constraints are increased by a factor of 5 or 10 depending on the relative size of the violation.

All of the penalty factor adjustment schemes successfully served their purposes of keeping the limit violations within the feasible region at the minimum solution point. Numerical instability can occur in any method when the minimum lies on one or more of the boundary constraints.

4.3 Comparison of Computer Results

Computer results for the minimum costs and losses optimal load flow solution were obtained for the IEEE 14-, 30- and 57-bus test systems[12]. Quadratic cost equation data was taken from reference [11] and is shown in Table 1. The cost at the minimum solution point for each case is shown in Table 2. All methods achieved reasonable accuracy in attaining the minimum cost; however, the steepest descent optimal load flow would tend to oscillate in the proximity of the minimum point. Therefore, the error criterion was relaxed from 10^{-4} to $.8 \cdot 10^{-3}$ per unit for that method.

A comparison of the execution time and number of iterations for each case is shown in Table 3. The times shown are for execution time only and do not include the compiler time. Clearly, steepest descent approaches the neighborhood of minimum point fastest, but if allowed to complete an additional iteration, will not satisfy the 10^{-4} tolerance described above for most cases tried. Therefore, these times should be weighed against the degree of accuracy desired. It is possible that the Hessian matrix method could achieve a very high degree of accuracy faster than steepest descent.

A comparison of the array storage for each case is shown in Table 4. These values reflect only the array storage and do not include the object code. Again, the steepest descent method requires the least amount of array storage for all of the cases tried. Sparsity techniques were not attempted because of the small size of these test systems compared with the large size of practical systems. However, it is doubtful that either of the other systems would be able to achieve less storage than steepest descent for these reasons.

i) In the steepest descent method the bus admittance and Jacobian matrices are used, which have the same relative sparsity.

ii) In the Fletcher-Powell optimal load flow, the inverse Hessian is formed, which is a large full matrix. In addition, several large, full matrices are required.

iii) The Hessian matrix optimal load flow uses the bus admittance and Jacobian matrices and, in addition, the large Hessian matrix is formed. This matrix is less sparse than the Jacobian due to coupling terms.

4.4 Alternative to Newton-Raphson Load Flow

From the results presented in the previous section, steepest descent appears most promising as an applied optimal load flow technique. However, the execution times for the cases tried indicate that improvements are necessary before on-line applications become popular. For these examples it was noted that the largest amount of execution time was spent in constructing and factoring the Jacobian matrix for repeated

Table 1. Quadratic Cost Equation Data for the Test Systems

System	Bus No.	c_0	c_1	c_2
14-bus				
	1	50.0	245.0	105.0
	2	50.0	351.0	44.4
	6	50.0	389.0	40.6
30-bus				
	1	50.0	245.0	105.0
	2	50.0	351.0	44.4
	11	50.0	389.0	40.6
57-bus				
	1	50.0	245.0	105.0
	3	50.0	389.0	44.0
	8	50.0	285.0	95.0
	12	50.0	351.0	40.6

$$f_i(P_i^G) = c_0(P_i^G)^2 + c_1 P_i^G + c_2$$

Table 2. Comparison of Final Costs at the Minimum Point

System	Steepest Descent	Fletcher-Powell	Hessian Matrix
	Cost ($/hr)	Cost ($/hr)	Cost ($/hr)
14-bus	761.29	760.87	760.50
30-bus	1,136.44	1,135.79	1,135.71
57-bus	6,524.72	6,523.54	6,523.62

Table 3. Comparison of Program Execution Times and Number of Iterations

System	Steepest Descent		Fletcher-Powell		Hessian Matrix	
	Execution Time (sec)	Iterations	Execution Time (sec)	Iterations	Execution Time (sec)	Iterations
14-bus	9.23	4	33.54	5	14.76	3
30-bus	38.13	4	114.79	4	52.44	3
57-bus	147.60	5	529.10	5	162.30	4

Table 4. Comparison of Program Array Storage

System	Steepest Descent Storage (K BYTE)	Fletcher-Powell Storage (K BYTE)	Hessian Matrix Storage (K BYTE)
14-bus	13.72	21.39	17.84
30-bus	31.64	52.41	39.75
57-bus	57.59	91.87	73.34

load flows. Some reduction of time is possible by reusing the factored Jacobian for a second voltage correction when nearing the solution. This technique could also be applied to the Hessian matrix optimal load flow with regard to reusing the Hessian matrix when nearing the minimum solution point.

Another technique has been employed which decouples real power and reactive power minimizations using the Fletcher-Powell method[8]. However that method was not found attractive from the studies performed here. It is also possible to decouple Newton-Raphson load flow equations[7]. Stott and Alsac developed an approach, called fast decoupled, to the Newton-Raphson load flow which neglects the off-diagonal Jacobian submatrices and approximates the remaining diagonal submatrices by constant matrices in equation (2.5). The voltage magnitude and angle corrections are then found separately in alternate iterations. This approach has been found to work well for most cases. It is identical with the Newton-Raphson method except for the evaluation of the Jacobian matrix, which is now constant and can be left in factored form.

The steepest descent optimal load flow program was modified by replacing the common load flow with the fast decoupled load flow (see Figure 6). Computer results were obtained for IEEE 14-, 30- and 57-bus systems. A comparison with the Newton-Raphson results is shown in Table 5. A vast improvement in the execution time of the steepest descent method was noted. Some additional array storage was required; however, since now both the Jacobian matrix and the decoupled, constant approximation must be stored.

Table 5. Comparison of Steepest Descent Optimal Load Flow Results

System	Newton-Raphson		Fast Decoupled	
	Execution Time (sec)	Array Storage (K BYTE)	Execution Time (sec)	Array Storage (K BYTE)
14-bus	9.23	13.72	4.68	15.86
30-bus	38.13	31.64	18.27	34.55
57-bus	147.60	57.59	65.98	63.84

4.5 Summary

It is clear from these results that steepest descent optimal load flow outweighs the other methods studied in most respects. Apart from slight convergence problems at the minimum, it ranked superior in all categories. Finally, the substitution of the fast decoupled load flow in the steepest descent optimal load flow holds much promise for further applications.

V. CONCLUSIONS

The conclusion of these studies focuses on the possible practical importance of the steepest descent optimal load flow. This method ranks high with regard to the other methods studied as far as execution time and array storage. The primary drawback of this method was observed to be the slight final convergence problem. This may be due in part to the penalty factor selection, as supposed by some[9]. For very small problems, it is possible to scale the variables so that the elliptical curves of Figure 1 become more circular and convergence is improved[16]. However, this approach is not practical for large systems with many variables and has not received much attention.

The substitution of the fast decoupled load flow in the steepest descent method greatly improved the execution time performance. There was some increase in the array storage; however, it is felt that this was heavily outweighed. The primary consideration for on-line applications is the computer execution time for repeated solutions. Presumably, in an actual application periodic solutions will be performed on-line whenever changes occur in the system's configuration or requirements. Since the load on a power system constantly experiences small changes in the demand power, the optimal load flow could be either executed at periodic intervals or when the mismatch between demand and supply became too great.

The optimal load flow in its present form is useful for system studies, especially planning, although some improvements must be made for on-line applications. Some recommendations for further studies in this area follow.

i) An improvement of the convergence characteristics of the steepest descent optimal load flow may be possible through the selection and adjustment of the penalty factors.

ii) An investigation into the scaling of the optimal load flow problem could yield some useful guidelines to improve the convergence of the steepest descent method.

iii) The improvement in execution times could be further increased and storage reduced if a decoupling procedure could be developed for the minimization routine with regard to the control variables.

There are many other possible areas of further study for the optimal load flow problem; for example, the methods of nonlinear optimization can be used to schedule other power systems variables, as well as plan future system requirements. It is hoped that through this work the methods of nonlinear programming gain more attention in the field of power system operation.

BIBLIOGRAPHY

1. H. W. Dommel and W. F. Tinney, "Optimal Power Flow Solutions", IEEE Trans. Power App. Syst., Vol. 87, pp. 1866-1876, October 1968.

2. A. M. Sasson, "Nonlinear Programming Solutions for Load-Flow, Minimum-Loss and Economic Dispatching Problems", IEEE Trans. Power App. Syst., Vol. 88, pp. 399-409, April 1969.

3. A. M. Sasson, "Combined Use of the Powell and Fletcher-Powell Nonlinear Programming Methods for Optimal Load Flow", IEEE Trans. Power App. Syst., Vol. 88, pp. 1530-1537, October 1969.

4. A. M. Sasson, F. Viloria, F. Aboytes, "Optimal Load Flow Solution Using the Hessian Matrix", IEEE Trans. Power App. Syst., Vol. 92, pp. 31-41, January 1973.

5. J. E. VanNess, "Iterative Methods for Digital Load Flow Studies", AIEE Trans. on P.A.S., Vol. PAS-78, pp. 583-588, August, 1959.

6. J. F. Dopazo, O. A. Klitin, G. W. Stagg and M. Watson, "An Optimization Technique for Real and Reactive Power Allocation", Proc. IEEE, Vol. 55, pp. 1877 - 1885, November 1967.

7. B. Stott and O. Alsac, "Fast Decoupled Load Flow", IEEE Trans. Power App. Syst., Vol. PAS-93, pp. 859-869, May/June, 1974.

8. R. Billinton and S. S. Sachdeva, "Real and Reactive Power Optimization by Suboptimum Techniques", IEEE Trans. Power App. Syst., Vol. 92, pp. 950-956, May 1973.

9. A. M. Sasson, F. Aboytes, R. Cardenas, F. Gomez, F. Viloria, "A Comparison of Power Systems Static Optimization Techniques", Proc. 7th PICA Conf., pp. 329-337, Boston, Mass., 1971.

10. A. M. Sasson and H. M. Merrill, "Some Applications of Optimization Techniques to Power Systems Problems", Proc. IEEE, Vol. 62, pp. 959-972, July 1974.

11. G. F. Reid and L. Hasdorff, "Economic Dispatch Using Quadratic Programming", IEEE Trans. Power App. Syst., Vol. 92, pp. 2015-2023, November 1973.

12. IEEE Standard Test Data, Courtesy of American Electric Power, New York, N.Y., 1962.

13. L. K. Kirchmayer, Economic Operation of Power Systems, John Wiley and Sons, Inc., 2nd Printing, 1967.

14. O. I. Elgerd, Electric Energy Systems Theory: An Introduction, McGraw-Hill Book Company, 1971.

BIBLIOGRAPHY, cont.

15. G. W. Stagg and A. H. El-Abiad, Computer Methods in Power System Analysis, McGraw-Hill Book Company, 1968.

16. D. G. Luenberger, Introduction to Linear and Nonlinear Programming, Addison-Wesley Publishing Company, 1973.

APPENDIX A

The Terms of the Jacobian Matrix in Polar Form

The terms of the Jacobian matrix in equation (2.5) are given below.

$$H_{kk} = \frac{\partial P_k}{\partial \delta_k} = -Q_k + V_k^2 Y_{kk} \sin\psi_{kk} \tag{A.1}$$

$$H_{ki} = \frac{\partial P_k}{\partial \delta_i} = V_k Y_{ki} V_i \sin(\psi_{ki} + \delta_k - \delta_i) \quad i \neq k \tag{A.2}$$

$$N_{kk} = V_k \frac{\partial P_k}{\partial V_k} = P_k + V_k^2 Y_{kk} \cos\psi_{kk} \tag{A.3}$$

$$N_{ki} = V_i \frac{\partial P_k}{\partial V_k} = V_k Y_{ki} V_i \cos(\psi_{ki} + \delta_k - \delta_i) \quad i \neq k \tag{A.4}$$

$$J_{kk} = \frac{\partial Q_k}{\partial \delta_k} = P_k - V_k^2 Y_{kk} \cos\psi_{kk} \tag{A.5}$$

$$J_{ki} = \frac{\partial Q_k}{\partial \delta_i} = -V_k Y_{ki} V_i \cos(\psi_{ki} + \delta_k - \delta_i) \quad i \neq k \tag{A.6}$$

$$L_{kk} = V_k \frac{\partial Q_k}{\partial V_k} = Q_k + V_k^2 Y_{kk} \sin\psi_{kk} \tag{A.7}$$

$$L_{ki} = V_k \frac{\partial Q_k}{\partial V_i} = V_k Y_{ki} V_i \sin(\psi_{ki} + \delta_k - \delta_i) \quad i \neq k \tag{A.8}$$

APPENDIX B

The Terms of the Gradient Vector for the Steepest Descent Optimal Load Flow

The terms of the gradient vector of equation (3.16) are given for the minimum costs and losses problem by the following equations.

$$V_i \frac{\partial L}{\partial V_i} = \frac{\partial f_s(P_s^G)}{\partial V_i} N_{si} + \Sigma_k \lambda_k^P N_{ki} + \Sigma_k \lambda_k^Q L_{ki} + \Sigma_k V_k \frac{\partial w_k(V_k)}{\partial V_k} \quad \text{(B.1)}$$

$$\frac{\partial L}{\partial P_s^G} = \frac{\partial f_s(P_s^G)}{\partial P_s^G} + \frac{\partial w_s(P_s^G)}{\partial P_s^G} \quad \text{(B.2)}$$

$$\underset{i \neq s}{\frac{\partial L}{\partial P_i^G}} = \frac{\partial f_i(P_i^G)}{\partial P_i^G} - \lambda_i^P \quad \text{(B.3)}$$

APPENDIX C

The Terms of the Gradient Vector for the Fletcher-Powell Optimal Load Flow

The terms of the gradient vector of equation (3.21) are given by the equations below.

$$\frac{\partial L}{\partial E_k} = \Sigma_i \frac{\partial f_i(P_i^G)}{\partial E_k} + \frac{\partial P_s}{\partial E_k} + \Sigma_i \frac{2(h_i(y)+s_i)}{r_i} \frac{\partial h_i}{\partial E_k} + \Sigma_j \frac{2(g_j(y)+s_j)}{r_j} \frac{\partial g_j}{\partial E_k} \quad (C.1)$$

$$\frac{\partial L}{\partial F_k} = \Sigma_i \frac{\partial f_i(P_i^G)}{\partial F_k} + \frac{\partial P_s}{\partial F_k} + \Sigma_i \frac{2(h_i(y)+s_i)}{r_i} \frac{\partial h_i}{\partial F_k} + \Sigma_j \frac{2(g_j(y)+s_j)}{r_j} \frac{\partial g_j}{\partial F_k} \quad (C.2)$$

The constraint terms in the above equations are given by the following terms.

$$\frac{\partial V_k}{\partial E_k} = \frac{E_k}{\sqrt{E_k^2 + F_k^2}} \quad (C.3)$$

$$\frac{\partial V_k}{\partial F_k} = \frac{F_k}{\sqrt{E_k^2 + F_k^2}} \quad (C.4)$$

$$\frac{\partial P_k^G}{\partial E_k} = 2G_{kk}E_k + \sum_{\substack{i=1 \\ i\neq k}}^{n} (G_{ki}E_i - B_{ki}F_i) \quad (C.5)$$

$$\frac{\partial P_i^G}{\partial E_k} = G_{ik}E_i + B_{ik}F_i \quad (C.6)$$

$i\neq k$

$$\frac{\partial P_k^G}{\partial F_k} = 2G_{kk}F_k + \sum_{\substack{i=1 \\ i\neq k}}^{n} (G_{ki}F_i + B_{ki}E_i) \quad (C.7)$$

$$\frac{\partial P_i^G}{\partial F_k}\Bigg|_{i\neq k} = G_{ik}F_i - B_{ik}E_i \tag{C.8}$$

$$\frac{\partial Q_k^G}{\partial E_k} = -2B_{kk}E_k - \sum_{\substack{i=1\\ i\neq k}}^{n} (G_{ki}F_i + B_{ki}E_i) \tag{C.9}$$

$$\frac{\partial Q_i^G}{\partial E_k}\Bigg|_{i\neq k} = G_{ik}F_i - B_{ik}E_i \tag{C.10}$$

$$\frac{\partial Q_k^G}{\partial F_k} = 2B_{kk}F_k - \sum_{\substack{i=1\\ i\neq k}}^{n} (G_{ki}E_i - B_{ki}F_i) \tag{C.11}$$

$$\frac{\partial Q_i^G}{\partial F_k}\Bigg|_{i\neq k} = -B_{ik}F_i - G_{ik}E_i \tag{C.12}$$

The vita has been removed from the scanned document

ABSTRACT

Power systems optimal load flow studies are performed for standard test cases using several methods of minimization. Comparisons of the solutions are made based on digital computer results. A general discussion is made of optimization and the necessary conditions for solution. Cost functions are written for the minimum costs and losses problem and then augmented to satisfy variable constraints. Through the use of common routines and format, sufficient basis for comparison of the several methods was established. Among the methods, steepest descent was chosen as most attractive and a further study was performed. Results of this are presented and conclusions drawn to improve the economical operation of our power system utilities.