origins in Kron's equivalent circuits, and owes its utility to the modern computer. Perhaps some day the authors or their successors will extend the circuits to include the effects of harmonics, but I agree that such a refinement is not required now.

The theory presented in the paper appears to me to be entirely sound, but I shall not attempt to understand it.

Since this new motor is truly unique and should have a wide use, it ought to be given a distinctive name. It is the custom in scientific circles to give the name of its discoverer to every new theory or device, so there is good precedent for naming this motor after one of those who pioneered it. The present authors both have three-syllable names, so these are not really suitable. Since C. A. Nickle was the first to propose a single-phase motor with a stepped air gap, and as I believe he was granted a patent on it, I suggest that this new motor be named the "Nickle Motor." This is quite euphonius, and it has the advantage of connoting a very low-cost device, which the new design certainly promises to be.

Doran D. Hershberger and **John L. Oldenkamp:** We wish to thank Mr. Alger for his kind remarks on our analysis of the motor. As was pointed out by Mr. Alger, the equivalent circuit of this single-winding motor can be obtained from Kron's generalized equivalent circuit.

The initial development and analysis was done without the analyses that had been contributed by others. The final analysis used the techniques attributed to Kron as outlined in Alger's *The Nature of Polyphase Induction Machines*.

The reference to the work of C. A. Nickle is interesting and is appreciated. Unfortunately, no written records of Nickle's accomplishments could be found: it is believed that he increased the air gap of the unshaded section of a shaded-pole hysteresis motor to equalize the flux densities of the shaded and unshaded sections.

Manuscript received March 14, 1968.

Optimal Power Flow Solutions

HERMANN W. DOMMEL, MEMBER, IEEE, AND WILLIAM F. TINNEY, SENIOR MEMBER, IEEE

Abstract—**A practical method is given for solving the power flow problem with control variables such as real and reactive power and transformer ratios automatically adjusted to minimize instantaneous costs or losses. The solution is feasible with respect to constraints on control variables and dependent variables such as load voltages, reactive sources, and tie line power angles. The method is based on power flow solution by Newton's method, a gradient adjustment algorithm for obtaining the minimum and penalty functions to account for dependent constraints. A test program solves problems of 500 nodes. Only a small extension of the power flow program is required to implement the method.**

I. Introduction

THE SOLUTION of power flow problems on digital computers has become standard practice. Among the input data which the user must specify are parameter values based on judgment (e.g., transformer tap settings). More elaborate programs adjust some of these control parameters in accordance with local criteria (e.g., maintaining a certain voltage magnitude by adjusting a transformer tap). The ultimate goal would be to adjust the control parameters in accordance with one global criterion instead of several local criteria. This may be done by defining an objective and finding its optimum (minimum or maximum); this is the problem of static optimization of a scalar objective function (also called cost function). Two cases are treated: 1) optimal real and reactive power flow (objective function = instantaneous operating costs, solution = exact economic dispatch) and 2) optimal reactive power flow (objective function = total system losses, solution = minimum losses).

The optimal real power flow has been solved with approximate loss formulas and more accurate methods have been proposed [2]–[5]. Approximate methods also exist for the optimal reactive power flow [6]–[9]. Recently attempts have been made to solve the optimal real and/or reactive power flow exactly [10], [11]. The general problem of optimal power flow subject to equality and inequality constraints was formulated in 1962 [12], and later extended [13]. Because very fast and accurate methods of power flow solution have evolved, it is now possible to solve the optimal power flow efficiently for large practical systems.

This paper reports progress resulting from a joint research effort by the Bonneville Power Administration (BPA) and Stanford Research Institute [10]. The approach consists of the solution of the power flow by Newton's method and the optimal adjustment of control parameters by the gradient method. Recently it has come to the authors attention that a very similar approach has been successfully used in the U.S.S.R. [14]–[16].

The ideas are developed step by step, starting with the solution of a feasible (nonoptimal) power flow, then the solution of an unconstrained optimal power flow, and finally the introduction of inequality constraints, first on control parameters and then on dependent variables. Matrices and vectors are distinguished from scalars by using brackets; vectors with the superscript T (for transpose) are rows; without T they are columns.

Paper 68 TP 96–PWR, recommended and approved by the Power System Engineering Committee of the IEEE Power Group for presentation at the IEEE Winter Power Meeting, New York, N.Y., January 28–February 2, 1968. Manuscript submitted September 13, 1967; made available for printing November 16, 1967.

The authors are with the Bonneville Power Administration, Portland, Ore. 97208.

II. Feasible Power Flow Solution

The power flow in a system of N nodes obeys N complex nodal equations:

$$V_k e^{-j\theta_k} \sum_{m=1}^{N} (G_{km} + jB_{km}) V_m e^{j\theta_m} = P_{\text{NET}k} - jQ_{\text{NET}k}, \quad k = 1, \cdots, N \quad (1)$$

where

V_k	voltage magnitude at node k
θ_k	voltage angle at node k
$G_{km} + jB_{km}$	element of nodal admittance matrix
$P_{\text{NET}k}, Q_{\text{NET}k}$	net real and reactive power entering node k.

Using the notation $P_k(V, \theta) - jQ_k(V, \theta)$ for the left-hand side, (1) can be written as $2N$ real equations:

$$P_k(V, \theta) - P_{\text{NET}k} = 0, \quad k = 1, \cdots, N \quad (2)$$

$$Q_k(V, \theta) - Q_{\text{NET}k} = 0, \quad k = 1, \cdots, N. \quad (3)$$

Each node is characterized by four variables, $P_{\text{NET}k}$, $Q_{\text{NET}k}$, V_k, θ_k, of which two are specified and the other two must be found. Depending upon which variables are specified (note that control parameters are regarded as specified in the power flow solution), the nodes can be divided into three types:

1) slack node with V, θ specified and $P_{\text{NET}}, Q_{\text{NET}}$ unknown (for convenience this shall always be node 1, also $\theta_1 = 0$ as reference),
2) P, Q-nodes with $P_{\text{NET}}, Q_{\text{NET}}$ specified and V, θ unknown,
3) P, V-nodes with P_{NET}, V specified and Q_{NET}, θ unknown.

Unknown values $P_{\text{NET}}, Q_{\text{NET}}$ are found directly from (2) and (3). Therefore, the basic problem is to find the unknown voltage magnitudes V and angles θ. Let $[x]$ be the vector of all unknown V and θ, and $[y]$ the vector of all specified variables,

$$[x] = \begin{bmatrix} V, \theta \text{ on each } P, Q\text{-node} \\ \theta \text{ on each } P, V\text{-node} \end{bmatrix} \quad [y] = \begin{bmatrix} V_1, \theta_1 \text{ on slack node} \\ P_{\text{NET}}, Q_{\text{NET}} \text{ on each } P, Q\text{-node} \\ P_{\text{NET}}, V \text{ on each } P, V\text{-node} \end{bmatrix}. \quad (4)$$

Newton's method (polar form) is used to find $[x]$. The algorithm is as follows [1]: first select a number of equations from (2) and (3) equal to the number of unknowns in $[x]$ to form the vector $[g]$.

$$[g(x, y)] = \begin{bmatrix} \text{eq.(2), eq.(3) for each } P, Q\text{-node} \\ \text{eq.(2) for each } P, V\text{-node} \end{bmatrix}. \quad (5)$$

Successive improvements for $[x]$,

$$[x^{(h+1)}] = [x^{(h)}] + [\Delta x]$$

are then found by solving the set of linear equations

$$\left[\frac{\partial g}{\partial x}(x^{(h)}, y)\right] \cdot [\Delta x] = -[g(x^{(h)}, y)] \quad (6)$$

where $[\partial g/\partial x]$ is the Jacobian matrix. It is square with the ith row being the partial derivatives of the ith equation in (5) with respect to all unknowns in $[x]$. The iteration starts with an initial guess $[x^{(0)}]$. Newton's method has been developed into a very fast algorithm by exploiting the sparsity of the Jacobian matrix through optimally ordered elimination and compressed storage schemes [1].

III. Optimal Power Flow Without Inequality Constraints

When real or reactive power is controllable, the values P_{NET}, Q_{NET} entering the power flow solution must be separated:

$$P_{\text{NET}k} = P_{Gk} - P_{Lk}, \quad Q_{\text{NET}k} = Q_{Gk} - Q_{Lk}$$

where

P_{Gk}, Q_{Gk}	controllable part of real and reactive power entering node k (sign positive when entering, because this part is usually generation)
P_{Lk}, Q_{Lk}	fixed part of real and reactive power leaving node k (sign positive when leaving, because this part is usually load).

The objective function f to be minimized, subject to the equality constraints (5), can now be defined.[1]

$$f = \sum K_i(P_{Gi})$$

(sum over all controllable power sources) in the case of optimal real and reactive power flow, where K_i is the production cost for the power P_{Gi}. The slack node must be included in the sum with $P_{G1} = P_1(V, \theta) + P_{L1}$ (if no costs were associated with the power at the slack node, the minimization process would try to assign all power to the slack node).

$$f = P_1(V, \theta)$$

in the case of optimal reactive power flow. Since $P_{\text{NET2}}, \cdots, P_{\text{NETN}}$ are fixed in this case, minimizing the real power P_1 is identical with minimizing total system losses.

$$f = \text{any other objective function which one may wish to define.}$$

Assume for the moment that some of the independent variables in $[y]$ can be varied disregarding inequality constraints. These adjustable variables are called control parameters. Vector $[y]$ can then be partitioned into a vector $[u]$ of control parameters and a vector $[p]$ of fixed parameters,

$$[y] = \begin{bmatrix} [u] \\ [p] \end{bmatrix}.$$

Control parameters may be voltage magnitudes on P, V-nodes, transformer tap ratios, real power P_G available for economic dispatch, etc. Fig. 1(b) shows the effect of control parameters V_1 and V_2 on the objective function (f = total system losses) for the simple network of Fig. 1(a) (two generators feeding one load). The plotted contours are the locus of constant f; thus $V_1 = 0.915$, $V_2 = 0.95$ produces the same losses of 20 MW as $V_1 = 1.14$, $V_2 = 1.05$. In the optimal power flow we seek that set of control parameters for which f takes on its minimum value. (Fig. 1(b) has no minimum for unconstrained control parameters.)

Using the classical optimization method of Lagrangian multipliers [17], the minimum of the function f, with $[u]$ as independent variables,

$$\min_{[u]} f(x, u) \quad (7)$$

[1] Since (5) does not include the equality constraints for the slack node, the real power at the slack node must be treated as a function $P_1(V, \theta)$.

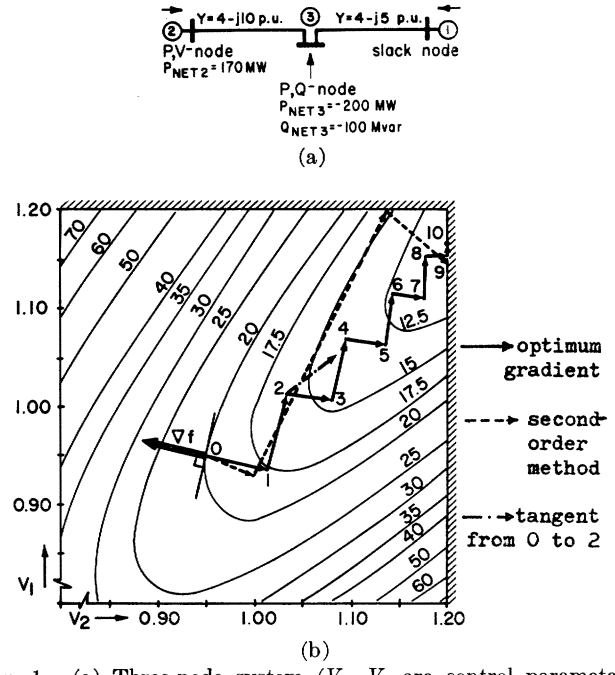

Fig. 1. (a) Three-node system (V_1, V_2 are control parameters). (b) Power flow solutions in V_1, V_2 space (contours = total system losses in MW; V_1, V_2 in per unit).

subject to equality constraints (5),

$$[g(x, u, p)] = 0 \tag{8}$$

is found by introducing as many auxiliary variables λ_i as there are equality constraints in (8) and minimizing the unconstrained Lagrangian function

$$\mathfrak{L}(x, u, p) = f(x, u) + [\lambda]^T \cdot [g(x, u, p)]. \tag{9}$$

The λ_i in $[\lambda]$ are called Lagrangian multipliers. From (9) follows the set of necessary conditions for a minimum:

$$\left[\frac{\partial \mathfrak{L}}{\partial x}\right] = \left[\frac{\partial f}{\partial x}\right] + \left[\frac{\partial g}{\partial x}\right]^T \cdot [\lambda] = 0 \tag{10}$$

$$\left[\frac{\partial \mathfrak{L}}{\partial u}\right] = \left[\frac{\partial f}{\partial u}\right] + \left[\frac{\partial g}{\partial u}\right]^T \cdot [\lambda] = 0 \tag{11}$$

$$\left[\frac{\partial \mathfrak{L}}{\partial \lambda}\right] = [g(x, u, p)] = 0$$

which is again (8). Note that (10) contains the transpose of the Jacobian matrix of the power flow solution (6) by Newton's method. For any feasible, but not yet optimal, power flow solution, (8) is satisfied and $[\lambda]$ can be obtained from (10). Then only $[\partial \mathfrak{L}/\partial u] \neq 0$ in (11). This vector has an important meaning; it is the gradient vector $[\nabla f]$, which is orthogonal to the contours of constant values of objective function (see Fig. 1(b) and Appendix I).

Equations (10), (11), and (8) are nonlinear and can only be solved by iteration. The simplest iteration scheme is the method of steepest descent (also called gradient method). The basic idea [18] is to move from one feasible solution point f in the direction of steepest descent (negative gradient) to a new feasible solution point with a lower value for the objective function (move to 1 in Fig. 1(b) starting from $V_1 = 0.95$, $V_2 = 0.95$). By repeating these moves in the direction of the negative gradient, the minimum will eventually be reached [moves to 2, 3, 4, $\cdots$ in Fig. 1(b)]. The solution algorithm for the gradient method is as follows.

1) Assume a set of control parameters $[u]$.

2) Find a feasible power flow solution by Newton's method. This yields the Jacobian matrix for the solution point in factored form (upper and lower triangular matrices), which is computationally equivalent to the inverse or transposed inverse [19].

3) Solve (10) for $[\lambda]$,

$$[\lambda] = -\left[\frac{\partial g}{\partial x}\right]^{T^{-1}} \cdot \left[\frac{\partial f}{\partial x}\right]. \tag{12}$$

This only amounts to one repeat solution of a linear system, for which the factored inverse is already available from step 2).

4) Insert $[\lambda]$ from (12) into (11) and compute the gradient

$$[\nabla f] = \left[\frac{\partial f}{\partial u}\right] + \left[\frac{\partial g}{\partial u}\right]^T \cdot [\lambda]. \tag{13}$$

The gradient $[\nabla f]$ measures the sensitivity of the objective function with respect to changes in $[u]$, subject to the equality constraints (8). Note that $[\partial f/\partial u]$ by itself does not give any helpful information because it ignores the equality constraints (8) of the power flow.

5) If $[\nabla f]$ is sufficiently small, the minimum has been reached.

6) Otherwise find a new set of control parameters from

$$[u^{\text{new}}] = [u^{\text{old}}] + [\Delta u] \text{ with } [\Delta u] = -c \cdot [\nabla f] \tag{14}$$

and return to step 2).

Steps 1) through 5) are straightforward and pose no computational problems. In the power flow solution [step 2)] one factorization of the Jacobian matrix can be saved when returning from step 6) by using the old Jacobian matrix from the previous adjustment cycle. This is justified when $[\Delta u]$ is not very large. After a few cycles, one repeat solution with the old Jacobian matrix plus one complete solution with a new Jacobian matrix are sufficient to give a new solution point.

The critical part of the algorithm is step 6). Equation (14) is one of several possible correction formulas (see Section VI). When (14) is used, much depends on the choice of the factor c. Too small a value assures convergence but causes too many adjustment cycles; too high a value causes oscillations around the minimum. In the so-called optimum gradient method the adjustment move is made to the lowest possible value of f along the given direction of the negative gradient (c variable in each cycle). The moves to 1, 2, 3, 4, $\cdots$ in Fig. 1(b) are those of the optimum gradient method.

The foregoing algorithm is based on the solution of the power flow by Newton's method. This choice was made because Newton's method has proven to be very efficient [1]; similar investigations in the U.S.S.R. [14], [16] seem to confirm this choice. However, the gradient can also be computed when the power flow is solved by other methods [11].

IV. Inequality Constraints on Control Parameters

In Section III it was assumed that the control parameters $[u]$ can assume any value. Actually the permissible values are constrained:

$$[u^{\min}] \leq [u] \leq [u^{\max}] \tag{15}$$

(e.g., $V^{\min} \leq V \leq V^{\max}$ on a P, V-node). These inequality constraints on control parameters can easily be handled by assuring that the adjustment algorithm in (14) does not send any parameter beyond its permissible limits. If the correction Δu_i from (14) would cause u_i to exceed one of its limits, u_i is set to the corresponding limit,

$$u_i^{new} = \begin{cases} u_i^{max}, & \text{if } u_i^{old} + \Delta u_i > u_i^{max} \\ u_i^{min}, & \text{if } u_i^{old} + \Delta u_i < u_i^{min} \\ u_i^{old} + \Delta u_i, & \text{otherwise.} \end{cases} \quad (16)$$

Even when a control parameter has reached its limit, its component in the gradient vector must still be computed in the following cycles because it might eventually back off from the limit.

In Fig. 1(b) the limit on V_2 is reached after the ninth cycle. In the tenth cycle the adjustment algorithm moves along the edge $V_2 = V_2^{max}$ into the minimum at $V_1 = 1.163$, $V_2 = 1.200$. When the limit of a parameter has been reached, the next move proceeds along the projection of the negative gradient onto the constraining equation $u_i = u_i^{max}$ or $u_i = u_i^{min}$; it is, therefore, called gradient projection technique. Since the projection is directly known in this case, its application is very simple (this is not the case for functional inequality constraints).

At the minimum the components $(\delta f/\delta u_i)$ of $[\nabla f]$ will be

$$\frac{\delta f}{\delta u_i} = 0, \quad \text{if } u_i^{min} < u_i < u_i^{max}$$

$$\frac{\delta f}{\delta u_i} \leq 0, \quad \text{if } u_i = u_i^{max} \quad (17)$$

$$\frac{\delta f}{\delta u_i} \geq 0, \quad \text{if } u_i = u_i^{min}.$$

The Kuhn–Tucker theorem proves that the conditions of (17) are necessary for a minimum, provided the functions involved are convex (see [10] and Appendix II).

The ability to handle parameter inequality constraints changes the usual role of the slack node. By making the voltage magnitude of the slack node a control parameter varying between V^{min} and V^{max} (normally with voltage magnitudes of some other nodes participating as control parameters) the slack no longer determines the voltage level throughout the system. Its only role is to take up that power balance which cannot be scheduled a priori because of unknown system losses.

Previously it has been shown [10], [20] that the Lagrangian multipliers and nonzero gradient components have a significant meaning at the optimum. The Lagrangian multipliers measure the sensitivity of the objective function with respect to consumption P_L, Q_L or generation P_G, Q_G and hence provide a rational basis for tariffication. The nonzero gradient components of control parameters at their limits measure the sensitivity of the objective function with respect to the limits u_i^{max} or u_i^{min} and consequently show the price being paid for imposing the limits or the savings obtainable by relaxing them.

V. Functional Inequality Constraints

Besides parameter inequality constraints on $[u]$ there can also be functional inequality constraints

$$h(x, u) \leq 0. \quad (18a)$$

Upper and lower limits on the dependent variables $[x]$ are a frequent case of functional constraints,

$$[x^{min}] \leq [x] \leq [x^{max}] \quad (18b)$$

where $[x]$ is a function of $[u]$ (e.g., $V^{min} \leq V \leq V^{max}$ on a P, Q-node). Fig. 2 shows the functional inequality constraint $V_3 \leq 1.0$ for the problem of Fig. 1. The end point 5 of the fifth move, from Fig. 1(b), already lies in the unfeasible region. If the functional constraints were accounted for by simply terminating the gradient moves at the constraint boundary, the process would

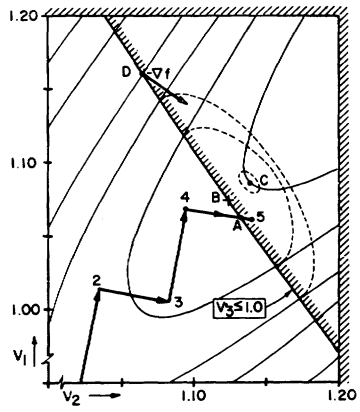

Fig. 2. Functional constraints.

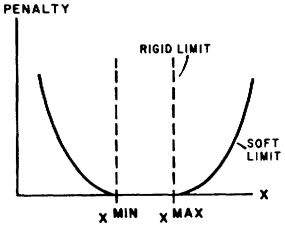

Fig. 3. Penalty function.

end at point A. However, the true minimum is at point B (here the difference in losses between A and B happens to be small; had point D been reached, the process would stop in D and the losses would be 20 MW compared with 12.9 MW in B).

Functional constraints are difficult to handle; the method can become very time consuming or practically impossible [21]. Basically a new direction, different from the negative gradient, must be found when confronting a functional constraint. It was proposed elsewhere [10] to linearize the problem when encountering such a boundary and to use linear programming techniques to find the new feasible direction. Another possibility is to transform the problem formulation, so that the difficult functional constraints become parameter constraints [21]; this approach is used in [16] by exchanging variables from $[x]$ into $[u]$ and vice versa. Another promising approach is the multiple gradient summation technique [27]. All three methods need the sensitivity matrix (Appendix I), or at least its dominant elements as an approximation, to relate changes in $[u]$ to changes in $[x]$.

Another approach is the penalty method [21]–[23] in which the objective function is augmented by penalties for functional constraint violations. This forces the solution back sufficiently close to the constraint. The penalty method was chosen for three reasons.

1) Functional constraints are seldom rigid limits in the strict mathematical sense but are, rather, soft limits (for instance, $V \leq 1.0$ on a P, Q-node really means V should not exceed 1.0 by too much, and $V = 1.01$ may still be permissible; the penalty method produces just such soft limits).

2) The penalty method adds very little to the algorithm, as it simply amounts to adding terms to $[\partial f/\partial x]$ (and also to $[\partial f/\partial u]$ if the functional constraint is also a function of $[u]$).

3) It produces feasible power flow solutions, with the penalties signaling the trouble spots, where poorly chosen rigid limits would exclude solutions (e.g., a long unloaded line with $P_{NET2} = Q_{NET2} = 0$ at the receiving end and voltage V_1 at the sending end controllable might have $V_2 > V_2^{max}$, even at the lowest setting $V_1 =$

$V_1^{\min}$. A rigid limit on V_2 excludes a solution, whereas the penalty method yields a feasible solution).

With the penalty method the objective function f must be replaced by

$$f^{\text{with penalties}} = f(x, u) + \sum w_j \tag{19}$$

where a penalty w_j is introduced for each violated functional constraint. On constraints (18b) the penalty functions used were

$$w_j = \begin{cases} s_j(x_j - x_j^{\max})^2, & \text{whenever } x_j > x_j^{\max} \\ s_j(x_j - x_j^{\min})^2, & \text{whenever } x_j < x_j^{\min}. \end{cases} \tag{20}$$

Fig. 3 shows this penalty function, which replaces the rigid limit by a soft limit. The steeper the penalty function (the higher the s_j) the closer the solution will stay within the rigid limit, but the worse the convergence will be. An effective method is to start with a low value for s_j and increase it during the optimization process if the solution exceeds a certain tolerance on the limit. The introduction of penalties into the objective function closes the contours (dashed lines in Fig. 2) and the minimum is then located in an unconstrained space. In the example of Figs. 1 and 2 with $V_3 \leq 1.0$, a penalty factor $s_3 = 7.5$ leads to point C ($V_3 = 1.016$) and raising the factor to $s_3 = 75.0$ leads to a point practically identical with B ($V_3 = 1.0015$).

On nodes with reactive power control, often two inequality constraints must be observed simultaneously,

$$V^{\min} \leq V \leq V^{\max} \tag{21}$$

$$Q^{\min} \leq Q_G \leq Q^{\max} \tag{22}$$

one of which is a parameter constraint and the other a functional constraint, depending on whether V or Q_G is chosen as control parameter. V was chosen as control parameter (with Q limits becoming functional constraints) for the following reasons.

1) In the power flow solution by Newton's method (polar form), only one equation enters for P, V-nodes (V = control parameter) versus two equations for P, Q-nodes (Q_G = control parameter).

2) The limits on V are more severe and thus more important, because they are physical limitations which cannot be expanded by technical means (the limits on Q_G can be expanded by installing additional reactive or capacitive equipment). As indicated in the first reason, the choice is influenced by the algorithm used; in [11] Q_G was used as control parameter. With V as control parameter, a violation of constraint (22) is best handled by introducing a penalty function. Another possibility is a change of the node type from P, V-node to P, Q-node, with Q_G becoming a control parameter; this transformation of the problem formulation is used in [16].

VI. Tests of Gradient Adjustment Algorithms

Test programs for the optimal power flow were written by modifying BPA's existing 500-node power flow program. A final version for production purposes is being programmed. Fig. 4 shows the simplified flow chart and Appendix III gives some computational details for the program. Various approaches for the critical adjustment algorithm were tested. The goal was to develop a method that would reach an acceptable near-optimum as fast as possible, rather than a method with extremely high accuracy at the expense of computer time. There is no need to determine the control parameters more accurately than they can be adjusted and measured in the actual system. Basically four versions were tested.

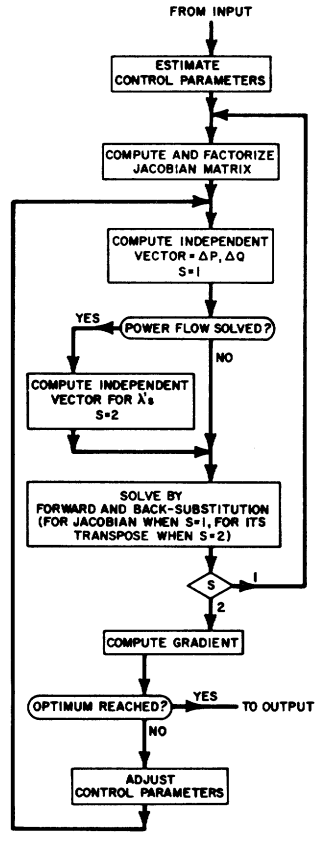

Fig. 4. Simplified flow chart.

1) *Second-Order Method Neglecting Interaction*

Approximate the objective function (19) by a quadratic function of $[u]$. Then the necessary conditions (first derivatives) become a set of linear equations, which can be solved directly for $[\Delta u]$,

$$[\Delta u] = -\left[\frac{\delta^2 f}{\delta u_i \delta u_k}\right]^{-1} \cdot [\nabla f] \tag{23}$$

where $[\delta^2 f/\delta u_i \delta u_k]$ is the Hessian matrix of second derivatives. If f were truly quadratic, (23) would give the final solution; otherwise, iterations are necessary. Finding the Hessian matrix and solving the set of linear equation (23) results in considerably more computer time per cycle than first-order gradient methods based on (14). It can also fail to converge if the Hessian matrix is not positive definite [18], where a first-order method might still converge. An approximate second-order method [10] neglects the off-diagonal elements in the Hessian matrix; this is justified when the control parameters have no (or little) interaction. In it the diagonal elements are found from a small exploratory displacement in $[u]$,

$$\frac{\delta^2 f}{\delta u_i^2} \approx \frac{\text{change in } (\delta f/\delta u_i)}{\text{change in } u_i}. \tag{24}$$

If any element $\delta^2 f/\delta u_i^2$ is negative, then the respective control parameter is left unchanged in that cycle, otherwise,

$$\Delta u_i = -\left(\frac{\delta^2 f}{\delta u_i^2}\right)^{-1} \cdot \left(\frac{\delta f}{\delta u_i}\right) \tag{25}$$

and u_i^{new} from (16). An example used by Smith and Tong [6] is well suited to illustrate the interaction problem; it is a loop

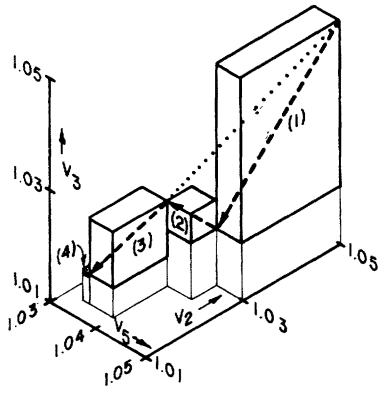

Fig. 5. Partan with three control parameters; (1) and (2) are optimum gradient moves, (3) is a tangent move, (4) is a solution point.

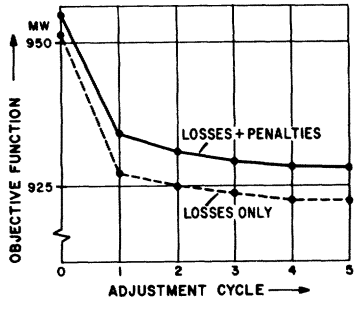

Fig. 6. Decrease in objective function.

system with the loop assumed open. When the loop is closed, the method is very successful; the two voltage adjustments seem to cause two mutually independent effects in the loop, with the total effect resulting from superposition. The method failed with the loop open; then the two voltages seem to interact considerably. The dashed line in Fig. 1(b) shows the performance of this method.

2) *Gradient and Optimum Gradient Method*

A solution can always be obtained by carefully choosing the factor c in (14). In the optimum gradient method, the factor c is chosen so that the minimum along the given direction is located (see Appendix III and Fig. 1(b)).

3) *Method of Parallel Tangents*

The gradient moves in Fig. 1(b) suggest that a considerable improvement can be made by moving in the direction of the tangent 0-2 after the first two gradient moves. This method has been generalized for the n-dimensional case under the name of Partan [24] (parallel tangents). The particular version best suited here is called steepest descent Partan in [24]. It performs well if there are not too many control parameters and if the contours are not too much distorted through the introduction of penalty functions. Fig. 5 shows the efficient performance of Partan for a five-node example taken from [11] with three control parameters (optimal reactive power flow with all five voltages as control parameters, of which two stay at the upper limit $V^{\max} = 1.05$, which was used as initial estimate). The first tangent move would end already close enough to the solution for practical purposes. It performed equally well on the open-loop system, Smith and Tong [6], where method 1) failed.

4) *Mixed Method*

A combination of methods 1) and 2) was finally adopted. Basically it uses the gradient method with the factor c chosen for the optimum gradient method or from simpler criteria (experiments are being made to find a satisfactory c without exploratory moves to save computer time). Whenever a gradient component changes sign from cycle $(h\text{-}1)$ to (h), its parameter is assumed to be close to the solution and (24) is used with

$$\frac{\delta^2 f}{\delta u_i^2} \approx \frac{\left(\frac{\delta f}{\delta u_i}\right)^{(h-1)} - \left(\frac{\delta f}{\delta u_i}\right)^{(h)}}{u_i^{(h-1)} - u_i^{(h)}} \tag{26}$$

provided $\delta^2 f/\delta u_i^2$ is positive. All examples used in the tests could be solved with this method.

Fig. 6 shows the decrease in the objective function for a realistic system with 328 nodes and 493 branches (80 parameters controllable). Note that most savings are realized in the first few cycles, which is highly desirable. Terminating the process after the fourth cycle resulted in a solution time of approximately four minutes (FORTRAN IV on IBM 7040); the Jacobian matrix was factored nine times and required 8400 words for its storage. With better programming the computer time could be reduced at least 50 percent.

VII. Future Improvements

Undoubtedly the methods outlined here can be further improved. Experiments to find the factor c faster in method 4) are being carried out. So far penalty functions have been used successfully to hold voltages V down close to $V^{\max}$ on P, Q-nodes and to hold tie line voltage angles close to specified values. More tests are planned for functional inequality constraints (22) and others.

Further improvements are possible, but very difficult to implement, through better scaling. A peculiarity of first-order gradient methods is that they are not invariant to scaling [25]. As an example assume that the contours of f are circles around the origin,

$$f(u_1, u_2) = u_1^2 + u_2^2$$

in which case the direction of steepest descent always points to the origin. If u_2 is scaled differently with $\bar{u}_2 = a \cdot u_2$, the contours for the new variables u_1, $\bar{u}_2$ become ellipses and the direction of steepest descent generally does not point to the origin anymore. Fortunately, the scaling problem has not been very serious in the cases run; the use of per unit quantities seems to establish reasonable scaling. In this context the second-order method (23) can be viewed as a gradient method with optimal scaling and rotation transformation.

Going from first-order to second-order methods (without neglecting interaction) could improve the convergence, but at a high price for additional computations. Therefore, it is doubtful whether the overall computer time would be cut down. If second-order methods are used, the inverse Hessian matrix could either be built up iteratively [26] or computed approximately by making small exploratory displacements Δu_i individually for each control parameter. In the latter case, it might be possible to use the same Hessian matrix through all cycles (if f were quadratic, the Hessian matrix would be constant). Second-order methods might be useful for on-line control, if the Hessian matrix is almost constant; it could then be precalculated and used unchanged as long as no major changes occur in the system.

VIII. Conclusions

It has been shown that Newton's method of power flow solution can be extended to yield an optimal power flow solution that is feasible with respect to all relevant inequality constraints. The main features of the method are a gradient procedure for finding the optimum and the use of penalty functions to handle functional inequality constraints. A test program that accommodates problems of 500 nodes has been written for the IBM 7040. Depending on the number of control variables, an optimal solution usually requires from 10 to 20 computations of the Jacobian matrix. The method is of importance for system planning and operation. Further improvements are expected.

Appendix I

Relationship Between Lagrangian Multipliers and Sensitivity Matrix

An alternate approach in computing the gradient uses a sensitivity matrix instead of Lagrangian multipliers as intermediate information. By definition the scalar total differential is

$$df = [\nabla f]^T \cdot [du] \tag{27}$$

or with $f = f(x, u)$

$$df = \left[\frac{\partial f}{\partial u}\right]^T \cdot [du] + \left[\frac{\partial f}{\partial x}\right]^T \cdot [dx]. \tag{28}$$

The dependent vector $[dx]$ in (28) can be expressed as a function of $[du]$ by expanding (8) into a Taylor series (with first-order terms only):

$$\left[\frac{\partial g}{\partial x}\right][dx] + \left[\frac{\partial g}{\partial u}\right][du] = 0$$

or

$$[dx] = [S][du] \tag{29}$$

where

$$[S] = -\left[\frac{\partial g}{\partial x}\right]^{-1} \cdot \left[\frac{\partial g}{\partial u}\right]. \tag{30}$$

$[S]$ is the sensitivity matrix. By inserting (29) into (28) and comparing it with (27), the gradient becomes

$$[\nabla f] = \left[\frac{\partial f}{\partial u}\right] + [S]^T \left[\frac{\partial f}{\partial x}\right]. \tag{31}$$

The amount of work involved in computing $[\nabla f]$ from (13) and (31) is basically the same; it is quite different, however, for finding the intermediate information. Computing the Lagrangian multipliers in (12) amounts to only one repeat solution of a system of linear equations, compared with M repeat solutions for the sensitivity matrix in (30). (M = number of control parameters.) Therefore, it is better to use the Lagrangian multipliers, provided the sensitivity matrix is not needed for other purposes (see Section V).

To show that $[\partial \mathcal{L}/\partial u]$ in (11) is the gradient $[\nabla f]$ when (8) and (10) are satisfied, insert (30) into (31):

$$[\nabla f] = \left[\frac{\partial f}{\partial u}\right] - \left[\frac{\partial g}{\partial u}\right]^T \cdot \left[\frac{\partial g}{\partial x}\right]^{T^{-1}} \cdot \left[\frac{\partial f}{\partial x}\right]. \tag{32}$$

This is identical with the expression in (11) after inserting $[\lambda]$ from (12), so that $[\partial \mathcal{L}/\partial u] = [\nabla f]$.

Appendix II

Kuhn–Tucker Formulation

The optimization problem with inequality constraints for the control parameters can be stated as

$$\min_{[u]} f(x, u) \tag{33}$$

subject to equality constraints

$$[g(x, u, p)] = 0 \tag{34}$$

and subject to inequality constraints

$$[u] - [u^{\max}] \leq 0 \tag{35}$$

$$[u^{\min}] - [u] \leq 0. \tag{36}$$

The Kuhn-Tucker theorem gives the necessary conditions (but no solution algorithm) for the minimum, assuming convexity for the functions (33)–(36), as

$$[\nabla \mathcal{L}] = 0 \quad \text{(gradient with respect to } u, x, \lambda) \tag{37}$$

and

$$\left.\begin{aligned} [\mu^{\max}]^T([u] - [u^{\max}]) &= 0 \\ [\mu^{\min}]^T([u^{\min}] - [u]) &= 0 \\ [\mu^{\max}] \geq 0, \quad [\mu^{\min}] &\geq 0 \end{aligned}\right\} \text{(exclusion equations).} \tag{38}$$

$\mathcal{L}$ is the Lagrangian function of (9) with additional terms μ to account for the inequality constraints:

$$\mathcal{L} = f(x, u) + [\lambda]^T[g(x, u, p)] + [\mu^{\max}]^T([u] - [u^{\max}]) + [\mu^{\min}]^T([u^{\min}] - [u]) \tag{39}$$

where $[\mu^{\max}]$ and $[\mu^{\min}]$ are the dual variables associated with the upper and lower limits; they are auxiliary variables similar to the Lagrangian multipliers for the equality constraints. If u_i reaches a limit, it will either be $u_i^{\max}$ or $u_i^{\min}$ and not both (otherwise u_i would be fixed and should be included in $[p]$); therefore, either inequality constraint (35) or (36) is active, that is, either $\mu^{\max}$ or $\mu^{\min}$ exists, but never both. Equation (37) becomes

$$\left[\frac{\partial \mathcal{L}}{\partial x}\right] = \left[\frac{\partial f}{\partial x}\right] + \left[\frac{\partial g}{\partial x}\right]^T [\lambda] = 0 \tag{40}$$

$$\left[\frac{\partial \mathcal{L}}{\partial u}\right] = \left[\frac{\partial f}{\partial u}\right] + \left[\frac{\partial g}{\partial u}\right]^T [\lambda] + [\mu] = 0 \tag{41}$$

where

$$\mu_i = \mu_i^{\max}, \quad \text{if } \mu_i > 0$$

$$\mu_i = -\mu_i^{\min}, \quad \text{if } u_i < 0$$

$$\left[\frac{\partial \mathcal{L}}{\partial \lambda}\right] = [g(x, u, p)] = 0. \tag{42}$$

The only difference with the necessary conditions for the unconstrained minimum of (10), (11), and (8) lies in the additional $[\mu]$ in (41). Comparing (41) with (13) shows that $[\mu]$, computed from (41) at any feasible (nonoptimal) power flow solution, with $[\lambda]$ from (40), is identical with the negative gradient. At the optimum, $[\mu]$ must also fulfill the exclusion equations (38), which say that

$$\mu_i = 0, \quad \text{if } u_i^{\min} < u_i < u_i^{\max}$$

$$\mu_i = \mu_i^{\max} \geq 0, \quad \text{if } u_i = u_i^{\max}$$

$$\mu_i = -\mu_i^{\min} \leq 0, \quad \text{if } u_i = u_i^{\min}$$

which is identical with (17) considering that $[\mu] = -[\nabla f]$.

Appendix III

Outline of the Computer Program

The sequence of computations is outlined in the simplified flow chart of Fig. 4. The part which essentially determines computer time and storage requirements is labeled "compute and factorize Jacobian matrix." It uses the algorithm from BPA's power flow program [1] and differs mainly in the additional storage of the lower triangular matrix (note that this involves no additional operations). The nonzero elements of the upper and transposed lower triangular matrices are stored in an interlocked array (element k-m of upper is followed by element m-k of lower triangular matrix); thus a shift in the starting address by ± 1 switches the algorithm from upper to lower triangular matrix or vice versa. This makes it easy to solve the system of linear equations either for the Jacobian matrix (in the power flow solution) or its transpose (in the λ computation). This shift is indicated by the switch S.

Power Flow Solution

The set of linear equations being solved in the power flow loop ($S = 1$) is

$$\begin{bmatrix} [H] & [N] \\ [J] & [L] \end{bmatrix} \cdot \begin{bmatrix} [\Delta\theta] \\ \left[\frac{\Delta V}{V}\right] \end{bmatrix} = \begin{bmatrix} [\Delta P] \\ [\Delta Q] \end{bmatrix} \tag{43}$$

where $[\Delta\theta]$, $[\Delta V/V]$ are the vectors of voltage angle and relative voltage magnitude corrections, and $[\Delta P]$, $[\Delta Q]$ are the vectors of power residuals with the components

$$\Delta P_k = P_{\text{NET}k} - P_k(V, \theta), \qquad \Delta Q_k = Q_{\text{NET}k} - Q_k(V, \theta)$$

and $[H]$, $[N]$, $[J]$, and $[L]$ are submatrices of the Jacobian matrix with the elements

$$\begin{aligned} H_{km} &= \frac{\partial P_k(V, \theta)}{\partial \theta_m}, & N_{km} &= \frac{\partial P_k(V, \theta)}{\partial V_m} V_m \\ J_{km} &= \frac{\partial Q_k(V, \theta)}{\partial \theta_m}, & L_{km} &= \frac{\partial Q_k(V, \theta)}{\partial V_m} V_m. \end{aligned} \tag{44}$$

Lagrangian Multipliers

Once the power flow is accurate enough, $S = 2$ switches the algorithm over to the solution of

$$\begin{bmatrix} [H] & [N] \\ [J] & [L] \end{bmatrix}^T \begin{bmatrix} [\lambda_P] \\ [\lambda_Q] \end{bmatrix} = -z \begin{bmatrix} [H_1] \\ [N_1] \end{bmatrix} - \begin{bmatrix} \left[\sum \frac{\partial w_j(\theta)}{\partial \theta}\right] \\ \left[\sum \frac{\partial w_j(V)}{\partial V} V\right] \end{bmatrix}. \tag{45}$$

The penalty terms w_j in (45) and hereafter enter only if the objective function has been augmented with penalty functions which depend on the variables indicated by the partial derivative. $[\lambda_P]$ and $[\lambda_Q]$ are subvectors of the Lagrangian multipliers associated with real and reactive power equality constraints, respectively, and

$$[H_1] = \left[\frac{\partial P_1(V, \theta)}{\partial \theta}\right], \qquad [N_1] = \left[\frac{\partial P_1(V, \theta)}{\partial V} V\right]$$

$$z = \begin{cases} \dfrac{\partial K_1}{\partial P_{G1}} & \text{in the case of optimal real and reactive power flow} \\ 1.0 & \text{in the case of optimal reactive power flow.} \end{cases}$$

Gradient Vector

After (45) has been solved, the gradient with respect to all control parameters is computed. Its components are as follows.

1) For voltage control:

$$\frac{\delta f}{\delta V_i} = \frac{1}{V_i}\left(zN_{1i} + \sum_{\substack{m = \text{all nodes} \\ \text{adjacent to and} \\ \text{including } i}} \lambda_{Pm} N_{mi} + \sum_{\substack{m = P,Q\text{-nodes} \\ \text{adjacent to } i}} \lambda_{Qm} L_{mi}\right) + \sum \frac{\partial w_j(V_i)}{\partial V_i}. \tag{46}$$

2) For power source control:

$$\begin{aligned} \frac{\delta f}{\delta P_{G1}} &= \frac{\partial K_1}{\partial P_{G1}} + \sum \frac{\partial w_j(P_{G1})}{\partial P_{G1}} \\ \frac{\delta f}{\delta P_{Gi}} &= \frac{\partial K_i}{\partial P_{Gi}} - \lambda_{Pi} + \sum \frac{\partial w_j(P_{Gi})}{\partial P_{Gi}}. \end{aligned} \tag{47}$$

3) For transformer tap control:

$$\frac{\delta f}{\delta t_{ik}} = \frac{1}{t_{ik}}(a_i N_{ik} + b_i H_{ik} + a_k N_{ki} + b_k H_{ki}) + 2V_k^2(b_k B_{ik} - a_k G_{ik}) + \sum \frac{\partial w_j(t_{ik})}{\partial t_{ik}} \tag{48}$$

where

$$a_i = \begin{cases} z, & \text{if } i = 1 \text{ (slack node)} \\ \lambda_{Pi}, & \text{otherwise} \end{cases}$$

$$b_i = \begin{cases} \lambda_{Qi}, & \text{if } i \text{ is a } P, Q\text{-node} \\ 0, & \text{otherwise} \end{cases}$$

analogous for a_k, b_k

$$G_{ik} + jB_{ik} = -t_{ik}Y_{ik}$$

t_{ik} = transformer turns ratio.

In (48) the transformer is assumed to enter the nodal admittance matrix with

	ith column	kth column
ith row	Y_{ik}	$-t_{ik}Y_{ik}$
kth row	$-t_{ik}Y_{ik}$	$t_{ik}^2 Y_{ik}$

where Y_{ik} is constant (if Y_{ik} is taken as a function of t_{ik}, then (48) must be modified). Changing transformer tap settings poses no problem since the Jacobian matrix is recalculated anyhow. A gradient component for phase shifting transformer control could be computed similarly.

Feasible Direction

With the gradient components from (46), (47), (48) or any other type of control parameter, the feasible direction of steepest descent $[r]$ is formed with

$$r_i = \begin{cases} 0, & \text{if } \dfrac{\delta f}{\delta u_i} < 0 \text{ and } u_i = u_i^{\max} \\ 0, & \text{if } \dfrac{\delta f}{\delta u_i} > 0 \text{ and } u_i = u_i^{\min} \\ -\dfrac{\delta f}{\delta u_i}, & \text{otherwise.} \end{cases} \tag{49}$$

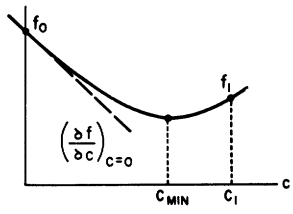

Fig. 7. Objective function along given direction.

Then the adjustments in this direction follow from

$$\Delta u_i = c \cdot r_i. \tag{50}$$

Optimum Gradient Method

The objective function f of (19) becomes a function of the scalar c only when the control parameters are moved in the direction of $[r]$. Let $f = f(c)$ be approximated by a parabola (Fig. 7). Then $c_{\min}$ for the minimum of f can be found from three values. One value f_0 is already known and a second value $\partial f/\partial c$ at $c = 0$ is readily calculated. Since by definition

$$\Delta f = f(c) - f_0 = \sum \frac{\delta f}{\delta u_i} \Delta u_i$$

$\partial f/\partial c$ at $c = 0$ becomes

$$\left(\frac{\partial f}{\partial c}\right)_{c=0} = -\sum r_i^2. \tag{51}$$

A third value f_1 is found from an exploratory move with a guessed $c = c_1$ (only power flow loop with $S = 1$ is involved). The final move is then made from c_1 to $c_{\min}$, where the gradient will be calculated anew for the next adjustment cycle. Some precautions are necessary because the actual curve differs from a parabola.

Instead of locating the minimum by an exploratory move, one could also construct the parabola solely from the information at the specific solution point ($c = 0$). Here one minimizes $\mathfrak{L}$ with respect to the scalar c

$$\frac{\partial \mathfrak{L}}{\partial c} = \frac{\partial f}{\partial c} + \left[\frac{\partial g}{\partial c}\right]^T [\lambda] = 0. \tag{52}$$

This is an equation for $c_{\min}$ whose value can be found by inserting $[u^{\text{new}}] = [u^{\text{old}}] + c[r]$. If $f(c)$ is assumed to be a parabola, the second and higher order terms for $c_{\min}$ are neglected in evaluating (52). This theoretical parabola was found to be less satisfactory than the experimental parabola.

References

[1] W. F. Tinney and C. E. Hart, "Power flow solution by Newton's method," *IEEE Trans. Power Apparatus and Systems*, vol. PAS-86, pp. 1449–1460, November 1967.

[2] R. B. Squires, "Economic dispatch of generation directly from power system voltages and admittances," *AIEE Trans. (Power Apparatus and Systems)*, vol. 79, pp. 1235–1245, February 1961.

[3] J. F. Calvert and T. W. Sze, "A new approach to loss minimization in electric power systems," *AIEE Trans. (Power Apparatus and Systems)*, vol. 76, pp. 1439–1446, February 1958.

[4] R. B. Shipley and M. Hochdorf, "Exact economic dispatch—digital computer solution, *AIEE Trans. (Power Apparatus and Systems)*, vol. 75, pp. 1147–1153, December 1956.

[5] H. Dommel, "Digital methods for power system analysis" (in German), *Arch. Elektrotech.*, vol. 48, pp. 41–68, February 1963 and pp. 118–132, April 1963.

[6] H. M. Smith and S. Y. Tong, "Minimizing power transmission losses by reactive-volt-ampere control," *IEEE Trans. Power Apparatus and Systems*, vol. 82, pp. 542–544, August 1963.

[7] R. Baumann, "Power flow solution with optimal reactive flow" (in German), *Arch. Elektrotech.*, vol. 48, pp. 213–224, 1963.

[8] K. Zollenkopf, "Tap settings on tap-changing transformers for minimal losses" (in German), *Elektrotech. Z.*, vol. 86, pt. A, pp. 590–595, 1965.

[9] F. Heilbronner, "The effect of tap-changing transformers on system losses" (in German), *Elektrotech. Z.*, vol. 87, pp. 685–689, 1966.

[10] J. Peschon, D. S. Piercy, W. F. Tinney, O. J. Tveit, and M. Cuenod, "Optimum control of reactive power flow," *IEEE Trans. Power Apparatus and Systems*, vol. PAS-87, pp. 40–48, January 1968.

[11] 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.

[12] J. Carpentier, "Contribution to the economic dispatch problem" (in French), *Bull. Soc. Franç. Elect.*, vol. 8, pp. 431–447, August 1962.

[13] G. Dauphin, G. D. Feingold, and D. Spohn, "Optimization methods for the power production in power systems" (in French), *Bulletin de la Direction des Etudes et Recherches, Electricité de France*, no. 1, 1966.

[14] L. A. Krumm, "Summary of the gradient method for optimizing the power flow in an interconnected power system" (in Russian), *Isv. Acad. Nauk USSR*, no. 3, pp. 3–16, 1965.

[15] K. A. Smirnov, "Optimization of the performance of a power system by the decreasing gradient method" (in Russian), *Isv. Akad. Nauk USSR*, no. 2, pp. 19–28, 1966.

[16] A. Z. Gamm, L. A. Krumm, and I. A. Sher, "Optimizing the power flow in a large power system by a gradient method with tearing into sub-systems" (in Russian), *Elektrichestvo*, no. 1, pp. 21–29, January 1967.

[17] G. Hadley, *Nonlinear and Dynamic Programming*. Reading, Mass.: Addison-Wesley, 1964, p. 60.

[18] H. A. Spang, "A review of minimization techniques for nonlinear functions," *SIAM Review*, vol. 4, pp. 343–365, October 1962.

[19] W. F. Tinney and J. W. Walker, "Direct solutions of sparse network equations by optimally ordered triangular factorization," *Proc. IEEE*, vol. 55, pp. 1801–1809, November 1967.

[20] J. Peschon, D. S. Piercy, W. F. Tinney, and O. J. Tveit, "Sensitivity in power systems," *IEEE Trans. Power Apparatus and Systems*, vol. PAS-87, pp. 1687–1696, August 1968.

[21] D. S. Grey, "Boundary conditions in optimization problems," in *Recent Advances in Optimization Techniques*, A. Lavi and T. P. Vogl, Eds. New York: Wiley, 1966, pp. 69–79.

[22] R. Schinzinger, "Optimization in electromagnetic system design," *Ibid.*, pp. 163–213.

[23] V. M. Gornshtein, "The conditions for optimization of power system operation with regard to operational constraints by means of penalty functions" (in Russian), *Elektrichestvo*, no. 8, pp. 39–44, 1965. Translation in Electric Technology U.S.S.R., pp. 411–425, 1965.

[24] B. V. Shah, R. J. Buehler, and O. Kempthorne, "Some algorithms for minimizing a function of several variables," *SIAM J.*, vol. 12, pp. 74–92, March 1964.

[25] D. J. Wilde, *Optimum Seeking Methods*. Englewood Cliffs, N.J.: Prentice-Hall, 1964, p. 119.

[26] R. Fletcher and M. J. D. Powell, "A rapidly convergent descent method for minimization," *Computer J.*, vol. 6, pp. 163–168, 1963.

[27] W. R. Klingman and D. M. Himmelblau, "Nonlinear programming with the aid of a multiple-gradient summation technique," *J. ACM*, vol. 11, pp. 400–415, October 1964.

Discussion

A.M. Sasson (Imperial College of Science and Technology, London England): The authors have presented an important contribution in the application of nonlinear optimization techniques to the load flow problem. The possibility of extending the techniques to other fields certainly should be of importance to a wide number of current investigators.

Manuscript received February 15, 1968.

The problem solved by the authors is the minimization of $f(x, u)$ subject to equality constraints, $g(x, u, p) = 0$, and inequality parameter and functional constraints. The recommended process is to satisfy the equality constraints by Newton's method, followed by the direct calculation of the Lagrangian multipliers and the minimization of the penalized objective function with respect to the control parameters. As the new values of the control parameters violate the equality constraints, the process has to be repeated until no further improvements are obtained. The comments of the authors are sought on the advantages or disadvantages of adding the equality constraints as penalty terms to the objective function. If this is done, the process would be reduced to the initial satisfaction of equality constraints by Newton's method to obtain a feasible nonoptimal starting point, followed by a minimization process which would require several steps, but which will always be approximately feasible. The gradient vector would have more terms as both x and u variables would be present.

Would the authors please clarify if the slack node is kept as voltage angle reference when its magnitude is a control parameter?

J. Peschon, J. C. Kaltenbach, and **L. Hajdu** (Stanford Research Institute, Menlo Park, Calif.): We are pleased to discuss this paper, since we were cooperating with the Bonneville Power Administration during the early phases of problem definition and search for practical computational methods. Being thus aware of the numerous and difficult problems the authors had to face before they accomplished their main goal—a reliable and efficient computer program capable of solving very high-dimensional power flow optimizations—we would like to congratulate them most heartily for their effort and to emphasize some other notable contributions contained in their paper.

They have introduced the notation

$$g(x, u) = 0$$

to describe the power flow equations, and they have identified the dependent variables x and the independent or control variables u. We are confident that this efficient notation will be retained by power system engineers, since it points out known facts that would be difficult to recognize with the conventional power flow notation.

A good illustration of this statement is contained in (10), from which it becomes clear that the computation of λ requires an inversion of the transposed Jacobian matrix. Since the inversion of the Jacobian matrix has already been performed in the power flow solution, this computation is trivial.

The authors give a detailed account, substantiated by experimental results of several gradient algorithms: second order without interaction, optimum gradient, and parallel tangents. Recognizing the fact that efficient gradient algorithms remain an art rather than a science, to be applied individually to each optimization problem, they have rendered a considerable service to the industry by demonstrating that the mixed method of second-order gradients and optimum gradients provides a good balance between speed of convergence and reliability of convergence.

Finally, they have shown that the penalty function method to account for inequality constraints on the dependent variables works well for the problem of power flow optimization. This again represents a considerable service to the industry, since penalty function methods sometimes work and sometimes do not. This fact can only be established experimentally, sometimes after months of programming agony.

To summarize these main points, we state that the authors have developed an efficient optimization method that can be implemented fairly readily once a good power flow program exists. They have shown, by experimentation and successive elimination of alternate gradient methods, that theirs represents the best compromise. For the problem stated, all of the technical and economic factors are taken into account, including the presence of variable ratio transformers.

Manuscript received February 16, 1968.

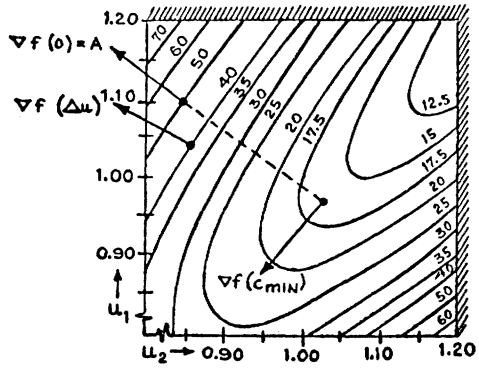

Fig. 8. Variation of the gradient ∇f with changes Δu in the vicinity of the original point.

Some may argue at this point that the power flow optimization problem stated is incomplete in the sense that certain important economical and technical factors are omitted, notably system reliability and vulnerability, cost of producing power in a mixed hydrothermal system or a system containing pumped storage, cost of thermal plant start up, and others. Their argument is correct but incomplete because the solution of a well-defined partial problem helps greatly toward the solution of an ill-defined or presently unsolvable global problem. A specific illustration of this statement is the economic optimization of mixed hydrothermal systems; once the value $\lambda_i(t)$ of power at the various nodes i of the system is known at various times t, the scheduling of hydroelectric production can be stated as a mathematical optimization problem, and solution methods can be developed. This fact was pointed out on the basis of intuitive considerations [28], it has also been demonstrated rigorously in the field of decomposition theory [29] where it is shown that the Lagrangian variables λ are interface variables capable of leading to a global problem solution by a sequence of subproblem optimizations, of which power flow optimization is one.

After these general comments, we would like to make a few specific remarks concerning the optimization procedure discussed.

The Hessian matrix $[\delta^2 f/\delta u_i\, \delta u_j]$ in (23) can be expressed explicitly in terms of the model equations f and g as follows [30]:

$$\left[\frac{\delta^2 f}{\delta u_i\, \delta u_j}\right] = \mathcal{L}_{uu} + S^T \mathcal{L}_{xx} S + 2S^T \mathcal{L}_{xu} \tag{53}$$

where the matrices $\mathcal{L}_{uu}$, $\mathcal{L}_{xx}$, and $\mathcal{L}_{xu}$ are the second partials of the function $\mathcal{L}$ in (9), and where the sensitivity matrix S is defined in (29). We wonder if the authors could comment on the difficulties of computing the elements of this matrix rather than obtaining its diagonal terms by exploration. Knowledge of this (and related second derivatives) is not only required by the theory of second-order gradients but is also highly desirable for determining the sensitivity properties of the cost function f with respect to sensing, telemetry, computation, and network model inaccuracies [30].

A closed-form algorithm for the optimum gradient may usefully supplement the experimental approaches summarized in (51) and (52). It proceeds as follows [31].

Let

$$\Delta f = A\, \Delta u + 1/2\, \Delta u^T B\, \Delta u \tag{54}$$

be the variation of cost with respect to changes Δu in the vicinity of the nominal point under discussion. The row vector A, of course, is the gradient ∇f of (13), and the symmetric matrix B is the Hessian matrix of (23). From (54), the gradient $\nabla f(\Delta u)$ can be expressed (see Fig. 8) in terms of Δu as

$$\nabla f(\Delta u) = A + \Delta u^T B. \tag{55}$$

Since the direction Δu is chosen along the original gradient $\nabla f(0) = A$ as

$$\Delta u = -cA^T \tag{56}$$

it follows that

$$\nabla f(c) = A - cAB. \tag{57}$$

The gradient directions $\nabla f(0)$ and $\nabla f(c)$ become perpendicular for $c_{\min}$ when the scalar product

$$\nabla f(c) \cdot \nabla f(0) = 0 \tag{58}$$

that is, when

$$c = c_{\min} = \frac{AA^T}{ABA^T}. \tag{59}$$

In Fig. 8, the optimum gradient method moves along the original gradient until the new gradient $\nabla f(c_{\min})$ and the original gradient are perpendicular, at which point no further cost reduction can be obtained along the original gradient direction.

Unlike the second-order adjustment, (23), the closed-form application of the optimum gradient method does not require an inversion of the Hessian matrix B. This development was given for the case of no constraints in control u: if a constraint of the type $u_i \leq u_i^{\max}$ or $u_i \geq u_i^{\min}$ has been encountered, the corresponding component of the gradient vector A is made zero, as is done in (49).

References

[28] J. Peschon, D. S. Piercy, O. J. Tveit, and D. J. Luenberger, "Optimum control of reactive power flow," Stanford Research Institute, Menlo Park, Calif., Bonneville Power Administration, February 1966.

[29] J. D. Pearson, "Decomposition, coordination, and multilevel systems," *IEEE Trans. Systems Science and Cybernetics*, vol. SSC-2, pp. 36–40, August 1966.

[30] J. Peschon, D. S. Piercy, W. F. Tinney, and O. J. Tveit, "Sensitivity in power systems," *PICA Proc.*, pp. 209–220, 1967.

[31] J. Carpentier, oral communication.

H. W. Dommel and **W. F. Tinney:** The authors are grateful for the excellent discussions which supplement the paper and raise questions which should stimulate further work in this direction.

Mr. Sasson asks about advantages or disadvantages in treating the power flow equality constraints as additional penalty terms rather than solving them directly. Since BPA's power flow program is very fast (about 11 seconds per Newton iteration for a 500-node problem on the IBM 7040) there was little incentive in this direction. After the first feasible solution in about three Newton iterations, one or two iterations usually suffice for another feasible solution with readjusted control parameters. Our experience with penalty terms for functional inequality constraints indicates that penalty terms usually distort the hypercontours in the state space and thus slow down the convergence. This has been particularly true with the method of parallel tangents. Therefore, it appears that one should use penalty terms only where absolutely necessary. However, this is not conclusive and Mr. Sasson's idea of treating the power flow equations as penalty terms is interesting enough to warrant further investigation. It might be a good approach in applications where not too much accuracy is needed for the power flow. Mr. Sasson is correct in assuming that the slack node is kept as voltage angle reference when its magnitude is a control parameter.

The authors fully agree with Messrs. Peschon, Kaltenbach, and Hajdu that the solution of a well-defined partial problem, here static optimization, is a prerequisite for attempts to solve global problems and welcome their comments about the significance of the values $\lambda_i(t)$ as interface variables.

The Hessian matrix in (53) is extremely difficult to compute for high-dimensional problems. In the first place, the derivatives $\mathcal{L}_{uu}$, $\mathcal{L}_{xx}$, $\mathcal{L}_{xu}$ involve three-dimensional arrays, e.g., in

$$\mathcal{L}_{xx} = \left[\frac{\partial^2 f}{\partial x^2}\right] + [\lambda]^T \left[\frac{\partial^2 g}{\partial x^2}\right]$$

where $[\partial^2 g/\partial x^2]$ is a three-dimensional matrix. This in itself is not the main obstacle, however, since these three-dimensional matrices are very sparse. This sparsity could probably be increased by rewriting the power flow equations in the form

$$\sum_{m=1}^{N} (G_{km} + jB_{km}) V_m e^{j\theta m} - \frac{P_{\text{NET}k} - jQ_{\text{NET}k}}{V_k e^{-j\theta k}} = 0$$

and applying Newton's method to its real and imaginary part, with rectangular, instead of polar, coordinates. Then most of the first derivatives would be constants [1] and, thus, the respective second derivatives would vanish. The computational difficulty lies in the sensitivity matrix $[S]$. To see the implications for the realistic system of Fig. 6 with 328 nodes, let 50 of the 80 control parameters be voltage magnitudes, and 30 be transformer tap settings. Then the sensitivity matrix would have 48 400 entries [605 × 80, where 605 reflects 327 P-equations (2) and 328 − 50 Q-equations (3)], which is far beyond the capability of our present computer. Aside from the severe storage requirements, which could be eased by storing and using dominant elements only, 80 repeat solutions would have to be performed (Appendix I). The computer time for this calculation would roughly be equivalent to ten adjustment cycles in the present method. Since about five cycles were enough for a satisfactory solution of this problem, the criterion of total computer time speaks for the present method. These difficulties in computing the Hessian matrix also make the closed-form algorithm derived in (54)–(59) impractical in spite of its theoretical elegance.

An alternate second-order method has been proposed by W. S. Meyer [32]. In his suggested approach, Newton's method is applied to the necessary conditions, (10), (11), and (8), with $[x]$, $[u]$ and $[\lambda]$ being simultaneous variables. The convergence behavior would be quadratic, and sparsity could be exploited. No moves from one to another feasible solution would have to be made since the power flow would not be solved until the very end of the entire optimization process.

The authors believe that their method has been proved to be practical for realistically large power systems. Improvements can be expected, of course, as more workers become interested in the optimal power flow problem, which embraces the entire constrained static optimization of all controllable power system parameters whether the application be economic dispatch, system planning, or something else.

References

[32] W. S. Meyer, personal communication.

Manuscript received March 14, 1968.