OPTIMAL POWER FLOW BY NEWTON APPROACH

David I. Sun, Member; Bruce Ashley, Member; Brian Brewer, Member; Art Hughes, Sr. Member
ESCA Corporation 13010 Northup Way Bellevue WA 98005

William F. Tinney, Fellow, Consultant

Abstract - The classical optimal power flow problem with a nonseparable objective function can be solved by an explicit Newton approach. Efficient, robust solutions can be obtained for problems of any practical size or kind. Solution effort is approximately proportional to network size, and is relatively independent of the number of controls or binding inequalities. The key idea is a direct simultaneous solution for all of the unknowns in the Lagrangian function on each iteration. Each iteration minimizes a quadratic approximation of the Lagrangian. For any given set of binding constraints the process converges to the Kuhn-Tucker conditions in a few iterations. The challenge in algorithm development is to efficiently identify the binding inequalities.

INTRODUCTION

This paper describes a new approach to the solution of the classical optimal power flow (OPF) problem based on an explicit Newton formulation. Tests performed on large problems with prototype algorithms show that the approach fulfills all of the requirements for practical OPF programs. The Newton formulation appears to be as fundamental and effective for OPF as it is for power flow (PF).

The classical OPF is a PF problem in which certain controllable variables are to be adjusted to minimize an objective function such as the cost of active power generation or losses, while satisfying physical and operating limits on various controls, dependent variables, and functions of variables. Because the objective must include losses, and the controls include reactive devices, the problem is characterized by a non-separable objective function. This characteristic, which sets the classical OPF apart from similar optimization problems, also makes it more difficult to solve.

The types of controls that an OPF must be able to accommodate include active and reactive power injections, generator voltages, transformer tap ratios, and phase shift angles. In a given OPF study, active power controls, reactive power controls, or a combination of both may be optimized. Because of the non-separability, optimizing reactive power control is more difficult than active power control. In general, any method that can solve the reactive problem can also solve the combined active-reactive problem. Therefore, the emphasis of this paper is on the reactive problem.

To be practical, OPF programs need to have performance requirements that include: a solution time that varies approximately in proportion to network size and is relatively independent of the number of controls or inequality constraints; rapid and consistent convergence to the Kuhn-Tucker (K-T) optimality conditions; absence of user supplied tuning and scaling factors for the optimization process; no compromises in OPF problem definitions; and problems of any practical size or kind should be as easily solvable by an OPF as by a PF. Newton-based OPF programs can satisfy these criteria for practicality.

The key idea of the approach is a sparsity-oriented, simultaneous solution for all of the unknowns of a quadratic approximation of the Lagrangian on each iteration. General large-scale nonlinear optimization problems are usually solved by quasi-Newton rather than explicit Newton methods since the latter would be too burdensome or even completely intractable. The OPF is a notable exception; the sparse Hessian matrix of the Lagrangian function can be explicitly evaluated and operated on efficiently. Efficiency can be further enhanced by decoupled formulations for which the convergence remains superlinear. For a given set of equalities, a Newton OPF converges to the K-T conditions in a few iterations. The major challenge in algorithm development is to identify the binding inequalities efficiently.

As described here, the Newton approach is a flexible formulation that can be used to develop different OPF algorithms suited to the requirements of different applications. Although the Newton approach exists as a concept entirely apart from any specific method of implementation, it would not be possible to develop practical OPF programs without employing special sparsity techniques. The concept and the techniques together comprise the given approach. Other Newton-based approaches are possible.

The paper describes the interim results of an ongoing research project, RP1724-1, being performed by ESCA Corporation under contract with the Electric Power Research Institute (EPRI). The purpose of the project is to explore OPF solution methods. More work has been done than is reported here and more work remains to be done before the project is completed. Other papers describing project results are planned.

NOTATION

The following symbol definitions are used throughout the text. Most symbols are also defined in the text where they first appear. Symbols used only in the APPENDIX are defined there.

Symbols

k, m	Subscripts denoting buses (nodes) or bus pairs.
Δ	Prefix on scalars and vectors denoting incremental correction of quantity.
wrt	"with respect to".
t	Superscript meaning transpose.

84 WM 044-4 A paper recommended and approved by the IEEE Power System Engineering Committee of the IEEE Power Engineering Society for presentation at the IEEE/PES 1984 Winter Meeting, Dallas, Texas, January 29 - February 3, 1984. Manuscript submitted September 1, 1983; made available for printing November 18, 1983.

∞	Matrix entry denoting a large number.

Scalars

P_k	Active power injection at bus k.
Q_k	Reactive power injection at bus k.
v_k	Magnitude of complex voltage at bus k.
θ_k	Angle of complex voltage at bus k.
t_{km}	Transformer tap ratio between buses k and m.
ϕ_{km}	Phase shifter angle between buses k and m.
y_k	Symbol for any state or control variable.
F	Objective function
L	Lagrangian function.
λ_{pk}	Lagrange multiplier for P_k.
λ_{qk}	Lagrange multiplier for Q_k.
μ_i	Lagrange multiplier for active inequality constraint i.
N	Number of buses in network.
S	Quadratic penalty weighting factor.
α_i	Quadratic penalty function for inequality constraint i.

Vectors

y	All variables v, θ, ϕ, t.
λ	All Lagrange multipliers λ_p and λ_q .
λ_p	Subvector of λ .
λ_q	Subvector of λ .
z	Composite of subvectors y and λ .
μ	Lagrange multipliers for binding inequality constraints.
g	Gradient of L wrt z.
z'	Subvector of z for real power variables.
z''	Subvector of z for reactive power variables.
g'	Subvector of g for real power variables.
g''	Subvector of g for reactive power variables.

Matrices

H	Hessian of the Lagrangian.
J	Jacobian for OPF.
W	Bordered Hessian. Composite of H, J, J^t.
W'	Bordered Hessian for Pθ subsystem.
W''	Bordered Hessian for Qv subsystem.
H'	Hessian submatrix of W'.
J'	Jacobian submatrix of W'.
H''	Hessian submatrix of W''.
J''	Jacobian submatrix of W''.

BACKGROUND

Progress on OPF analysis has been reviewed periodically [1-3], and it was reviewed for this project. Attempts to solve the OPF problem date back over twenty years. Practical solutions for OPF problems with separable objective functions have been obtained with special linear programming methods [4], but the classical OPF has defied practical solutions. Of the many proposed methods, only a few have been tested on problems large enough to evaluate their performance. It seems that all of these methods fall short of being practical because of limitations in speed, problem size or robustness, or because of compromises with the problem definition. This conclusion is supported by the observation that the classical OPF has not become a standard application.

Rather than attempt to evaluate many different methods, only three are discussed. They are relevant to the Newton approach and are representative of others.

The reduced gradient method of Dommel and Tinney [5] has been frequently cited as a benchmark. Several other reduced gradient methods have also been published. Although some success has been claimed for these methods, recent findings show that gradient methods cannot solve the OPF. This was convincingly shown in a recent paper by Burchett, Happ and Wirgau [6] in which they compared a reduced gradient method with a quasi-Newton method which is strong enough to solve the problem.

Sasson, Viloria and Aboytes [7] were the first to show that a sparse factorization of an explicit Hessian matrix for the OPF could be performed. However, they attempted to solve the PF equations as well as to minimize the objective function by using only the Hessian matrix in an augmented Lagrangian formulation. Evidently, this does not work well. With their formulation, the Hessian matrix had second-neighbor fill-in, making it considerably less sparse than the Hessian matrix of the approach described in this paper.

The aforementioned quasi-Newton method of Burchett, et al [6] utilizes second order information contained in an iteratively constructed reduced Hessian matrix. It is strong enough to solve the OPF. In each iteration the quasi-Newton method obtains a descent direction by operating on the reduced gradient with an approximation of the factors of the symmetrical but dense reduced Hessian. (The reduced Hessian has the dimension of the superbasic variables). Updating the dense approximate factors in each iteration, and the number of iterations required for the approximate Hessian to generate a good descent direction, contribute to a large computation and storage requirement for problems of practical size.

PRELIMINARIES

This section briefly describes how the basic aspects of nonlinear optimization are applied in the given approach.

Example Problem

The five bus network of Fig. 1 is used throughout as a specific example.

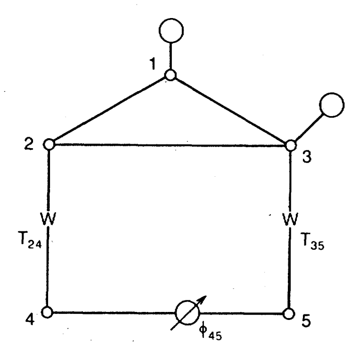


Fig. 1 Example Network

Generators at buses 1 and 3 are dispatchable sources of active and reactive power. Load buses 2, 4, and 5 have scheduled active and reactive power. The phase shifter has controllable angle ϕ_{45} and the transformers have controllable tap ratios t_{24} and t_{35}. The reference angle of the complex bus voltages is θ_1.

The problem is to minimize the total cost of P_1 and P_3 while satisfying the scheduled loads and limits of different variables. P_1, Q_1, P_3, Q_3, ϕ_{45}, t_{24} and t_{35} are controllable.

Variables

In an optimization procedure the variables can change roles. Variables are classified as control (superbasic), dependent state (basic) or, if they have reached their limits, constant (nonbasic). In the Newton approach these distinctions are unnecessary. Basic and superbasic variables are processed identically. Variables that become nonbasic are constrained at their limiting values without actually changing their roles as variables. The original variables establish the dimension of the vector of variables, y, and it is never changed.

In the example, y consists of thirteen variables; ϕ_{45}, t_{24}, t_{35}, θ_1, v_1, θ_2, v_2, θ_3, v_3, θ_4, v_4, θ_5, and v_5. Although θ_1, the reference angle, remains constant, it is included in y for completeness.

Inequality Constraints

The following quantities have upper and/or lower limits:

1. Dispatchable sources of P and Q.
2. Variables; voltages, tap ratios, phase shifter angles.
3. Functions such as line flows, measures of security, etc.

Active Equalities

The set of equalities, A, always includes the PF equations for scheduled load and generation. It also includes the following set of binding inequalities:

1. The PF equations of any dispatchable sources of P or Q that are constrained at their limits.
2. The equations of any other inequality functions constrained at their limits.
3. The trivial equations of variables constrained at their limits.

Objective Function

The objective function, F, can assume several different forms, but the differences have no significant impact on the approach. F for the example is given in (1).

$$F = C_1P_1 + C_3P_3 \tag{1}$$

C_1 and C_3 are the slopes of piecewise linear segments of the active power cost curves of dispatchable generators 1 and 3. In an actual program suitable logic would be needed to change C_1 and C_3 as P_1 and P_3 change in the solution process. This logic is omitted here because it has no impact on the approach. If desired, quadratic approximations of the cost curves could also be used in F.

Lagrangian Function

The Lagrangian, L, for the Newton OPF is shown in (2).

$$L = C_1P_1 + C_3P_3 - \Sigma_k\lambda_{pk}P_k - \Sigma_k\lambda_{qk}Q_k \tag{2}$$

The summations are over all buses from 1 to N. However, the Lagrange multipliers, λ_{q1} and λ_{q3}, become nonzero only when their respective equations for Q_1 and Q_3 are in A. The values of the Lagrange multipliers are determined by the solution process. In the five bus example, λ_{q1}, and λ_{q3} are zero because they correspond to dispatchable sources operating within their limits.

In (2), L is a linear combination of PF equations. In the Newton approach inequality constraints on variables and functions are enforced by quadratic penalty functions α_i which augment L as shown in (3).

$$L = C_1P_1 + C_3P_3 - \Sigma_k\lambda_{pk}P_k - \Sigma_k\lambda_{qk}Q_k - \Sigma_i\alpha_i \tag{3}$$

The summation for i is over all binding inequalities.

L for the example, with no active penalties and omitting Q_1 and Q_3 for dispatchable VAR sources, is,

$$\begin{aligned}L = {} & C_1P_1 + C_3P_3 - \lambda_{p1}P_1 - \lambda_{p2}P_2 - \lambda_{p3}P_3 \\ & - \lambda_{p4}P_4 - \lambda_{p5}P_5 - \lambda_{q2}Q_2 - \lambda_{q4}Q_4 - \lambda_{q5}Q_5\end{aligned} \tag{4}$$

In some optimization methods L is augmented by other equations to enhance its positive definiteness. Such augmentation, which adversely affects sparsity, is unnecessary in the Newton OPF.

Solution Conditions

A minimum of the objective function occurs when the K-T optimality conditions are satisfied. The quantitative indicators for evaluating the K-T conditions require no additional computation. The following conditions must exist for a minimum:

1. The mismatches of all PF equations in set A are within tolerance.
2. The inequality constraints are all satisfied.
3. The projected gradient is zero (except for roundoff).
4. The sensitivity, μ_i, between each binding inequality constraint and the objective function is such that further cost reduction can be achieved only if the constraint is violated.
5. The projection of the Hessian in the feasible region is positive definite. Conceptually, this means that the multi-dimensional objective function is bowl shaped; therefore, the stationary point is a true minimum, not a saddle point.

Although local, positive definiteness ensures that a stationary point is a minimum, it does not ensure that it is a global minimum.

SUBMATRICES

This section describes the Jacobian and Hessian matrices, J and H, and displays them for the example problem. J and H are submatrices in the Newton formulation. Their arrangements in the displays in this section are conceptual only. In an actual implementation they are combined and rearranged into a single matrix or matrix pair. However, the arrangement shown for H would be correct if its separate factorization were needed.

Jacobian Matrix

J is a matrix of first partial derivatives of PF equations wrt y. In the Newton approach the number of rows in J is always 2N and the number of columns is equal to the dimension of y. J for the example is shown is (5).

$J =$

	ϕ_{45}	T_{24}	T_{35}	θ_1	v_1	θ_2	v_2	θ_3	v_3	θ_4	v_4	θ_5	v_5
P_1				j	j	j	j	j	j				
Q_1				j	j	j	j	j	j				
P_2		j			j	j	j	j	j	j	j		
Q_2		j			j	j	j	j	j	j	j		
P_3			j		j	j	j	j	j			j	j
Q_3			j		j	j	j	j	j			j	j
P_4	j	j				j	j			j	j	j	j
Q_4	j	j				j	j			j	j	j	j
P_5	j		j					j	j	j	j	j	j
Q_5	j		j					j	j	j	j	j	j

(5)

The rows for currently inactive PF equations can be omitted from J. Each symbol j indicates a nonzero element and its coordinates indicate a specific derivative. For example, j with coordinates Q_4, v_5 is $\partial Q_4/\partial v_5$. A nonsingular Jacobian matrix for the Newton PF could be formed by deleting certain columns of J.

$H =$

	ϕ_{45}	T_{24}	T_{35}	θ_1	v_1	θ_2	v_2	θ_3	v_3	θ_4	v_4	θ_5	v_5
ϕ_{45}	h									h	h	h	h
T_{24}		h				h	h			h	h		
T_{35}			h					h	h			h	h
θ_1				∞									
v_1					h	h	h	h	h				
θ_2						h	h	h	h	h	h		
v_2							h	h	h	h	h		
θ_3								h	h			h	h
v_3									h			h	h
θ_4										h	h	h	h
v_4											h	h	h
θ_5												h	h
v_5													h

(6)

Hessian Matrix

H is a symmetric matrix of second partial derivatives of L wrt y. Its dimension is the same as that of y. Each element of H is a linear combination of second partial derivatives of PF equations. Details on forming H are given in the APPENDIX. H for the example is shown in (6).

Each symbol h indicates a nonzero element and its coordinates indicate a specific derivative. For example, h with coordinates t_{24}, v_4 is $\partial^2 L/\partial t_{24}\partial v_4$. Only the upper triangle is shown.

The sparsity of the 2x2 block structure of the main submatrix (heavy border) is the same as that of the network incidence matrix. Therefore, it can be factorized in the usual sparsity-directed way. Processing of elements outside of the heavy border can be effectively separated from the processing of the main submatrix. This will be clarified in the next section. H by itself, however, is not factorized in the Newton approach.

Elements of H representing the couplings between variable pairs θ and v, θ and t, and ϕ and v are very small compared to the average magnitudes of other elements. This favorable property is exploited in the decoupled formulations.

NEWTON OPF FORMULATIONS

In this section the coupled and decoupled Newton OPF formulations are explained by temporarily ignoring the complications of identifying and enforcing the binding inequalities. The only active constraints assumed at this point are the PF equations for the given loads. (Without some additional constraints the problem would be unsolvable since the optimal solution would normally be far outside any normal operating range.)

				1				2				3				4				5				Δz		-g
	h															h	h	j	j	h	h	j	j	$\Delta\phi_{45}$		$-\partial L/\partial\phi_{45}$
		h						h	h	j	j					h	h	j	j					ΔT_{24}		$-\partial L/\partial T_{24}$
			h									h	h	j						h	h	j	j	ΔT_{35}		$-\partial L/\partial T_{35}$
1				∞																				—		—
					h	j		h	h	j	j	h	h	j										Δv_1		$-\partial L/\partial v_1$
						∞		j	j			j	j											$\Delta\lambda_{p1}$		$-\infty\cdot\frac{\partial L}{\partial\lambda_{p1}}$
							∞																	—		—
2								h	h	j	j	h	h	j		h	h	j	j					$\Delta\theta_2$		$-\partial L/\partial\theta_2$
									h	j	j	h	h	j		h	h	j	j					Δv_2		$-\partial L/\partial v_2$
												j	j			j	j							$\Delta\lambda_{p2}$		$-\partial L/\partial\lambda_{p2}$
												j	j			j	j							$\Delta\lambda_{q2}$	=	$-\partial L/\partial\lambda_{q2}$
3												h	h	j						h	h	j	j	$\Delta\theta_3$		$-\partial L/\partial\theta_3$
													h	j						h	h	j	j	Δv_3		$-\partial L/\partial v_3$
														∞						j	j			$\Delta\lambda_{p3}$		$-\infty\cdot\frac{\partial L}{\partial\lambda_{p3}}$
															∞									—		—
4																h	h	j	j	h	h	j	j	$\Delta\theta_4$		$-\partial L/\partial\theta_4$
																	h	j	j	h	h	j	j	Δv_4		$-\partial L/\partial v_4$
																				j	j			$\Delta\lambda_{p4}$		$-\partial L/\partial\lambda_{p4}$
																				j	j			$\Delta\lambda_{q4}$		$-\partial L/\partial\lambda_{q4}$
5																				h	h	j	j	$\Delta\theta_5$		$\partial L/\partial\theta_5$
																					h	j	j	Δv_5		$-\partial L/\partial v_5$
																								$\Delta\lambda_{p5}$		$-\partial L/\partial\lambda_{p5}$
																								$\Delta\lambda_{q5}$		$-\partial L/\partial\lambda_{q5}$

(9)

Coupled Formulation

Let z be a vector composed of subvectors y and λ. Differentiating L twice wrt z leads to the symmetric matrix W symbolized in (7).

$$W = \begin{bmatrix} H & -J^t \\ -J & 0 \end{bmatrix} \qquad (7)$$

This matrix is well known (pg. 239 of ref. (12) but apparently unnamed. Here it is referred to simply as W. W provides the basis for the Newton OPF formulation shown in (8),

$$W \times \Delta z = -g \qquad (8)$$

where g is the gradient vector of first partial derivatives of L wrt z, and Δz is a vector of Newton corrections in z.

For sparse factorization it is necessary to rearrange W. Equation (9) on the previous page shows the sparsity arrangement for the example.

Equation (9) also shows in detail the elements that comprise the system. First derivatives of L wrt λ in vector g are the familiar residuals of the PF equations, i.e., the differences between the actual and scheduled injections of P_k and Q_k.

This arrangement creates a main submatrix (heavy borders) whose 4x4 block structure is the same as that of the network incidence matrix. Processing of the external submatrices can be effectively separated from processing of the main submatrix because elimination of the external variables only modifies certain nonzero blocks within the main submatrix. Furthermore, the ordering of the external variables has no effect on sparsity. (These same remarks also apply to the arrangement of H in (6).)

Factorization and repeat solution of W requires four times as much computational effort as the same operations with the power flow Jacobian, and the matrix storage requirement is double. Efficiency and storage requirements can be further improved by developing decoupled formulations.

Coupled Solution

The coupled OPF, subject only to equality constraints, can be solved by the following simplified algorithm;

1. Make starting estimates for z = (y,λ).

   y can be the same as the starting values for a PF. λ can be zero or any reasonable guess.

2. Evaluate g as a function of z.

3. Check for solution.

   If K-T conditions are satisfied, OPF is solved.

   Else

4. Evaluate W as a function of z.

5. Factorize W and solve for Δz.

6. Update z by Δz.

7. Return to step 2 with updated z.

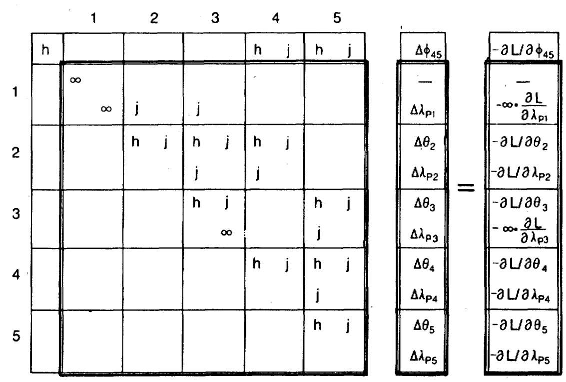
Decoupled Formulation

A decoupled formulation of the Newton OPF is symbolized in (10),

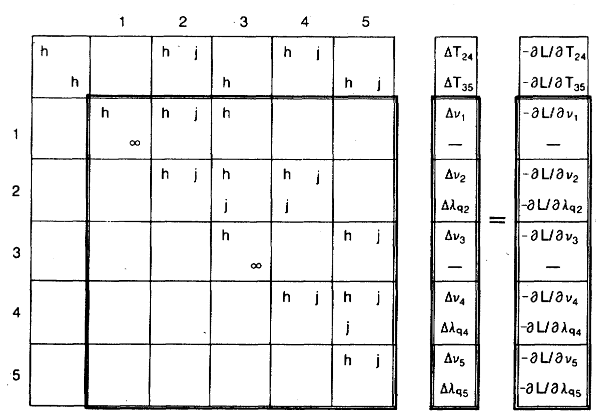
$$W' \times \Delta z' = -g' \qquad (10a)$$

$$W'' \times \Delta z'' = -g'' \qquad (10b)$$

where the symbols are as defined in the NOTATION section. The primes on symbols are used to suggest an analogy with the decoupled PF [9]. The sparsity arrangement of (10) for the example is shown in (11).



(11a)



(11b)

The decoupling shown divides the problem into Pθ and Qv sysbystems. The elements of H that are omitted in the decoupling are negligibly small; the elements of J that are omitted, which are the same as those omitted in the decoupled PF [9], may not be negligible in some problems. Any defects that might arise from this particular decoupling would be due to the approximation for J, not H. There are remedies for defects due to decoupling of J. Other versions of decoupling are possible. For example, versions based on decoupling H but not J.

The sparsity of the 2x2 block structure of the main submatrix of W′ and W″ (heavy borders) is the same as that of the network incidence matrix. The remarks about processing of the external submatrices for the coupled version also apply to this decoupled version.

Factorization of W′ and W″ together requires approximately the same computational effort as factorization of the Newton PF. The matrix storage is also similar.

Decoupled Solution

The decoupled OPF, subject only to equality constraints, can be solved by the following algorithm:

1. Make starting estimates for z.
2. Evaluate g' and W' as functions of z.
3. Factorize W' and solve for $\Delta z'$.
4. Update z by $\Delta z'$.
5. Evaluate g" as function of updated z.
6. Check for solution.

   If K-T conditions are satisfied, OPF is solved.

   Else
7. Evaluate W" as function of updated z.
8. Factorize W" and solve for $\Delta z''$.
9. Update z by $\Delta z''$.
10. Return to step 2 with updated z.

ENFORCING INEQUALITY CONSTRAINTS

This section discusses methods for enforcing inequality constraints. Methods for determining which inequalities to enforce are discussed in the next section. The following attributes are advantageous in inequality constraint enforcement methods:

1. The method should be efficient for enforcing inequalities singly or a few at a time without refactorizing.

2. The method should provide for automatic control of the hardness of constraint enforcement. There are situations where a certain amount of freedom around an inequality limit is beneficial. Certain strategies for identifying the binding inequalities require control of the hardness of enforcement. Soft limits may also be needed to obtain useful nonfeasible solutions when feasible ones do not exist.

3. The method should not make uncoordinated step changes in the variables or functions in order to set them to desired values. All such changes should be made in the simultaneous solution of the correction vector Δz. Uncoordinated changes disrupt convergence.

Inequality constraints are considered in three categories: (1) Dispatchable active and reactive power, (2) variables, (3) functions of variables.

Enforcing Limits on Dispatchable Power Sources

The PF equations for dispatchable VAR sources are omitted from A as long as they are feasible, and restored to A when necessary to prevent the Q_k from becoming infeasible. Adding and removing these equations causes no structural changes in W′ and W". As an example, suppose that at some iteration dispatchable reactive source at bus k exceeds its maximum. Then the equation for $\Delta\lambda_{qk}$ would be activated by entering the current evaluation of its elements in its previously dummied row/column in W". Since Q_k had been feasible up to this point, λ_{qk} would be currently zero, and this is the value that would be used in the initial evaluation. After the next iteration, λ_{qk} would assume a nonzero value. The sign of λ_{qk} indicates whether Q_k would return to its feasible range if its equation were again removed.

The PF equations themselves provide only hard enforcement. Methods for soft enforcement of limits on power sources have been developed but their explanation is beyond the scope of this paper.

Enforcing Limits on Variables

Any variable can be made into a constant by simply eliminating the equation for its correction and setting the variable to the desired value. For example, assume voltage at load bus k is below its minimum. The limit could be enforced by setting v_k to its limit and making row/column for v_k in W" a dummy. The equation for Q_k remains in W" to enforce its scheduled value.
The change of variables to constants, or vice versa, can be made in this manner whenever W′ or W" is refactorized, but more efficient methods are needed for enforcing one or a few inequalities at a time. Also, as pointed out, uncoordinated changes in variables should not be made. A variable should be moved to its relevant limit in coordination with the simultaneous correction of all other variables.

Quadratic penalty functions ideally fulfill the requirements for inequality constraint enforcement in second-order methods. This is in marked contrast to the erratic and generally unsuccessful behavior of linearized quadratic penalties used in gradient-based OPF methods. In a second-order method the effect of a quadratic penalty is accurately controllable. It provides a two-sided constraint that can clamp a variable at an exact target value or allow a controllable amount of freedom for variation around the target value. It also provides the means for moving a variable or function to the target value in coordination with the simultaneous correction of all other variables. Equation (12) shows the quadratic penalty function used to constrain variable y_i.

$$\alpha_i = \frac{S_i}{2}(y_i - \bar{y}_i)^2 \qquad (12)$$

where $\bar{y}_i$ is the target value, y_i is the current value and S_i is a weighting factor that is automatically controlled to give the appropriate amount of hardness of enforcement. The first and second derivatives of α_i are :

$$\frac{d\alpha_i}{dy_i} = S_i\,(y_i - \bar{y}_i) \qquad (13)$$

$$\frac{d^2\alpha_i}{dy_i^2} = S_i \qquad (14)$$

Application of the penalty will force y_i to coincide with or remain as close to $\bar{y}_i$ as desired. The quadratic penalty augments the L. Its first derivative is added to $\partial L/\partial y_i$ of g and its second derivative is added to $\partial^2 L/\partial y_i^2$ of W.

The proper value for S_i is automatically controlled. If S_i is large, $\bar{y}_i$ acts as a hard limit and y_i moves arbitrarily close to the limit. If S_i is small, $\bar{y}_i$ acts as a soft limit. The corrections in y_i are coordinated with all other corrections on each iteration. The value of $\partial^2 L/\partial y_i^2$ in W′ or W" is used in automatically adjusting S_i to the proper value for a soft limit. Any S_i larger than a certain value will effectively produce a hard limit.

After a penalty has been imposed to constrain a variable, the sign of its derivative can be used to decide whether the penalty is still needed. The derivative of a penalty on a variable is the negative of its Lagrange multiplier, μ_i, for the enforced limiting value. The derivatives of imposed penalties are also used to determine whether the K-T conditions are satisfied.

Quadratic penalties eliminate uncoordinated changes in variables, permit control of the hardness of constraint enforcement, indicate whether imposed constraints are still binding, and provide the incremental costs of the imposed constraints.

Enforcing Limits on Special Functions

It is necessary to be able to enforce limits on special functions. One example is the function defining the flow of power through a line. Limits on such functions could be enforced, when needed, by adding their equations explicitly to the Lagrangian. But this would be undesirable because it would radically change the favorable and otherwise constant sparsity structures of W′ and W". It is much simpler and equally effective to enforce such functional inequalities with penalties. The following example of an inequality constraint on the flow of power through a line illustrates the general case. Other functions defining inequalities can be handled in a similar way.

The power flow through line (km) can be held approximately within a specified limit by requiring the power angle $(\theta_k - \theta_m)$ to be close to some precomputed value $\bar{\theta}_{km}$. If this is not accurate enough, θ_{km} can be iteratively adjusted to achieve any desired degree of accuracy. The quadratic penalty function α_i for enforcing an angle limit $\bar{\theta}_{km}$ is:

$$\alpha_i = \frac{S_i}{2} (\theta_k - \theta_m - \bar{\theta}_{km})^2 \tag{15}$$

where S_i is a weighting factor.

The first and second derivatives of α_i are given in (16).

$$\frac{d\alpha_i}{d\theta_k} = S_i (\theta_k - \theta_m) \tag{16a}$$

$$\frac{d\alpha_i}{d\theta_m} = -S_i (\theta_k - \theta_m) \tag{16b}$$

$$\frac{d^2\alpha_i}{d\theta_k^2} = S_i \tag{16c}$$

$$\frac{d^2\alpha_i}{d\theta_m^2} = S_i \tag{16d}$$

$$\frac{d^2\alpha_i}{d\theta_m d\theta_k} = \frac{d^2\alpha_i}{d\theta_k d\theta_m} = -S_i \tag{16e}$$

To activate a penalty for $(\theta_k - \theta_m)$ evaluate the first derivatives of α_i and add them to the corresponding derivatives in g. Add $+S_i$ to $\partial^2 L/\partial\theta_k^2$ and $\partial^2 L/\partial\theta_m^2$, and add $-S_i$ to $\partial^2 L/\partial\theta_k\partial\theta_m$ of W′. Note that the penalty only augments existing non-zero terms in W′. Any number of penalties such as this can be imposed with no burden on the matrix. S_i can be adjusted automatically to suit requirements. The first derivative (the two are equal except for sign) is the Lagrange multiplier, μ_i, for the enforced inequality. It indicates whether it is necessary to continue enforcing the inequality, and it is also used in determining whether the K-T conditions are satisfied.

Individual Enforcement of Inequalities

The enforcement or removal of a single inequality constraint involves only a few small changes in W′ or W" and these changes affect only a few rows of the factorization. There are two sparsity methods for obtaining repeat solutions for small changes in a factorized matrix without having to perform a complete refactorization: partial refactorization and compensation. Either or both can be used to impose or remove inequality constraints singly or a few at a time without refactorizing W′ or W".

Partial refactorization exploits the property that small changes in a sparse matrix affect only a few rows of its sparse factorization. For simple changes, which occur most frequently, a special factor updating scheme [10] is most efficient. For more complicated changes the normal factorization is used but applied only to the affected rows. To impose or remove a PF equation or a penalty for an inequality, proceed as follows:

1. Modify the affected rows of the factors of W′ or W" by partial refactorization.
2. Modify g′ or g" by adding or removing the derivatives for the penalty or equation change.
3. Perform a fast forward solution [11] using only the rows of W′ or W" affected by the updating.
4. Back solve to obtain the updated solution.

Small matrix changes followed by updated solutions can be performed by this scheme repeatedly. The updates in the matrix and solutions are cumulative. The efficiency of each new update is unaffected by the preceding updates.

With compensation [11], which is similar to the Schur complement technique [13] used in optimization literature, the factors are not altered. The effect of matrix changes on the solution is computed directly from sparse compensation vectors. Successive matrix changes are not cumulative; each new solution by compensation is a change from the original base solution. However, successive changes can be accumulated in the compensation vectors to make solutions for cumulative changes more efficient. The efficiency of the method declines as the number of successive changes increases. Compensation is more efficient than partial refactorization up to a crossover point that depends on the total number and type of changes.

DETERMINING THE BINDING INEQUALITY SET

In decoupled schemes the Pθ and Qv subsystems are solved in alternating cycles. Some of the most effective algorithms tested thus far perform fast trial iterations within each cycle to determine the currently binding inequalities. Depending on the application, the scheme of trial iteration used in one cycle may differ from the one used in the alternate cycle. Schemes may also differ depending on the stage of the solution process. In the following simplified descriptions the particular cycle is not specified. Each algorithm could be used for either the Pθ or Qv cycle. Thus, instead of specifying W′ or W", the symbol W is used to mean either or both. The descriptions are incomplete; the intent is only to suggest strategies for finding the binding inequalities.

The algorithms described employ two kinds of solutions of the decoupled quadratic approximations of L: main iterations and trial iterations.

A main iteration for either cycle is as follows:

1. Evaluate W and g as functions of the latest update of z and A.

2. Factorize W and solve for Δz.

A trial iteration is performed either by partial refactorization or by compensation. The aim of trial iterations is to identify the currently binding inequalities. In each trial the effects of different combinations of constraint enforcement and release at a tentative solution point are examined. Various strategies are employed to minimize the number of trials needed to find the binding set. When the currently binding inequalities have been satisfactorily identified, the solution is advanced to the next step with the binding inequalities enforced. A trial iteration is always fast compared to a main iteration, but the ratio of their speeds depends on the number and kinds of matrix changes involved in the trial and the amount of monitoring required to examine the tentative solution.

Example Algorithms

The aim of this section is to suggest how algorithms based on the approach can be developed. Three rather fundamental algorithms, each of which could have many variations, are outlined. Other different algorithms are also possible.

Algorithm I

1. Check for solution. If not solved, continue.

2. Check inequalities.

   a. Change A for enforcement of all violated inequalities.

   b. Change A to release all previously enforced inequalities where enforcement is no longer needed.

3. Perform main iteration to obtain Δz.

4. Update z by Δz.

5. Proceed to alternate cycle.

Algorithm I is analogous to the schemes used in conventional PF programs to identify and enforce inequality constraints. It does not maintain strict feasibility but allows constraints to be violated before enforcing them on the following iteration. Constraints are enforced and released simultaneously in groups. As observed in PF programs, simultaneous schemes succeed in identifying the binding inequalities but they take many iterations. Algorithm I could be improved by starting with soft penalties and systematically hardening them as the binding set becomes clarified. Many different simultaneous enforcement strategies are possible.

Algorithm II

1. - 3. Same as in Algorithm I.

4. Find scalar K ($K \leq 1.0$) such that at $z + K(\Delta z)$ only one inequality has moved to its limit.

5. Update z by $K(\Delta z)$.

6. Proceed to alternate half cycle.

Except for the effect of the alternate decoupled cycle, Algorithm II would maintain strict feasibility. It would, however, be prohibitively slow. It could be greatly improved by adjusting K to allow for a controlled amount of violation on each iteration. A merit function [12] could be developed to determine a good value for K by taking into account the kind, number and magnitude of the violations. Many variations of the merit function and the strategy for its application are possible.

Algorithm III

1. - 3. Same as in Algorithm I.

4. Tentatively update z by Δz.

5. Check inequalities at $z + \Delta z$.

   a. If the inequalities satisfy the criteria for an acceptable iteration, advance the solution to the tentative step; return to step (1) and proceed to alternate half cycle.

   Else

   b. According to the trial strategy select a set of violated inequalities for enforcement and a set of previously enforced inequalities for release. Modify A according to the selection.

   Continue.

6. Enforce and release the selected inequalities by a trial iteration to obtain $\Delta z(\text{trial})$.

7. Return to step (5) with $z = \Delta z(\text{trial})$.

Algorithm III maintains a specified amount of feasibility. At the end of each series of trial iterations following a main iteration the enforcement of inequalities is exactly or approximately resolved depending on the criteria for an acceptable iteration. In effect, the trial iterations test different combinations of enforcement in order to find the best one. Many variations of Algorithm III are possible.

Criteria For Enforcing Inequalities

The enforcement of different types of inequalities has different effects. The effects are approximately predictable from the known behavior of power systems and the solution process. This knowledge can be used in establishing criteria for enforcing inequalities and in developing merit functions.

When a violated voltage limit at a load bus is enforced, it tends to produce changes in the same direction in other voltages in its area of the network. Simultaneous enforcement of hard limits on several violated voltage inequalities in one area can create disruptive reactive power flows. Enforcing voltage inequalities one at a time, or several at a time with soft constraints, alleviates this difficulty. Criteria can be developed for selecting the inequality most likely to be binding out of a group of voltage inequality violations.

Enforcing an inequality on a dispatchable source of active or reactive power forces the remaining dispatchable sources to make up for the deficiency or surplus. This action can force other dispatchable sources of the same kind to exceed their limits. These cascading effects should be resolved by trial iterations before proceeding. Violations of inequalities in this class should be enforced simultaneously because they generally tend to create more violations of the same kind.

Exact enforcement of all inequalities on each iteration is generally wasteful. This waste can be reduced by developing criteria for acceptance of an iteration that takes into account the number, kind and magnitude of violations as well as bus mismatches. These few examples of criteria are illustrative, but not exhaustive.

Selective Monitoring

The forward solutions for trial iterations in Algorithm III can be performed efficiently because only a few columns of the factors are directly involved. The back solutions can be speeded up in a similar way by solving only for those quantities that need to be monitored. After each trial iteration it is only necessary to check two groups of inequalities; (1) those that were nonfeasible on the last trial; and (2) those that were marginally feasible but threatened with violation by the enforcement of other inequalities. The first group would normally consist of bus voltages for which it would be necessary to check the effect of enforcing one on reducing the violations of the others. This group would rapidly diminish on each successive trial iteration. Since most of the effort of trial iterations is in the back solution, large gains can be made by selective monitoring.

TEST RESULTS

At the time of this writing the development and testing of algorithms is continuing. The intent of the investigation is to determine how different algorithms perform on different problems, not to produce a definitive solution method. All testing is being done on the VAX 11/780 computer. To give some evidence of how the approach works, the results of one experimental algorithm on one test problem are shown. This example is representative of results being obtained with different problems and different experimental algorithms. It is expected that much better algorithms will be developed.

Test Problem

The network of the test problem is a portion of the power system of the Northeastern United States. The problem is reactive control optimization only. Normally, multiple sources of active power can be dispatched. In this particular example, however, a single slack is used. The main attributes of the test problem are:

- Network: 912 buses. 1637 branches.
- Total MW generation: 1200 p.u. (100 MVA base)
- Starting Condition: unconverged network solution.
- Objective Function: Minimum active power losses
- Controls: 255 dispatchable reactive sources and 41 controllable transformers.
- Inequality Constraints: Upper and lower limits on dispatchable reactive sources, controllable transformer tap ratios and all bus voltages.

A version of Algorithm III for a decoupled formulation was used for the test. The strategy of inequality constraint enforcement was as follows: On the first iteration only the limits on generator voltages and transformer tap ratios were enforced. On the second iteration limits on reactive sources were also enforced. On the third and subsequent iterations all inequalities were enforced. All enforcements were with hard limits. Each Qv cycle was made completely feasible by trial iterations before the algorithm was advanced to the next cycle.

Results of Test Problem

The history of iterations for the test problem is shown in TABLE I. Each iteration actually represents both a Pθ and a Qv cycle. However, the trial iterations and inequality constraint enforcements apply only to the Qv cycle.

TABLE I.
Optimization Summary For A 912-Bus System

(1) Main Iter.	(2) No.of Trial Iter.	(3) No. of Bind'g Limits	(4)* Active Power Loss	(5)* Max P Msmtch	(6)* Max Q Msmtch	(7)* RMS of Projec'd Gradient	(8) Max ΔV or Δtap
1	11	142	11.448	0.286	6.136	.005554	.1824
2	8	143	10.325	0.418	0.820	.001167	.0689
3	23	131	10.271	0.303	0.344	.001417	.0462
4	2	133	10.302	0.097	0.126	.001367	.0186
5	1	132	10.306	0.024	0.028	.000253	.0193
6	2	131	10.316	0.048	0.036	.000366	.0225
7	0	131	10.327	0.066	0.041	.000318	.0128
8	0	131	10.332	0.062	0.014	.000094	.0027
9	0	131	10.334	0.048	0.004	.000047	.0006
10	0	131	10.331	0.030	0.004	.000046	.0002
11	0	131	10.333	0.014	0.004	.000043	.0001
12	0	131	10.334	0.002	0.002	.000034	.0001

* 100 MVA base.

Legend For Table I

Column 1 is the count of iterations. An iteration consists of one main iteration for the Pθ cycle and one main iteration plus a variable number of trial iterations for the Qv cycle. The system is updated at the end of the main iteration of the Pθ cycle. For the Qv cycle the system is not updated until the end of the trial iterations.

Column 2 is the count of trial iterations needed to identify the currently binding inequalities for an acceptable iteration. If the main iteration itself is feasible, no trials are needed and the count is zero.

Column 3 is the number of binding inequalities at the end of the iteration.

Column 4 is the value of the objective function at the start of the iteration.

Column 5 and 6 are the largest residuals of the PF equations at the start of the iteration.

Column 7 is the RMS of the gradient of L for the basic and superbasic variables at the beginning of the iteration. The RMS value is the square root of the sum of the squared gradients, divided by the number of the basic and superbasic variables. This number is a measure of the steepness of the objective function. At the solution it should be zero except for roundoff.

Column 8 is the maximum per unit iterative correction in a voltage or transformer tap ratio. At the solution it should be zero except for roundoff.

DISCUSSION OF RESULTS

The total computational effort for the solution can be estimated from the total of main and trial iterations. For a well-written code each main iteration would require about 1.2 times as much effort as a Newton PF iteration. The effort of trial iterations is variable. With good sparsity techniques the cost of a trial should average about one tenth of a full iteration. This makes the total effort for the test problem equivalent to about 18 Newton PF iterations.

Most of the trial iterations were performed at an early stage. Eleven trials were needed in the first iteration to find the currently binding taps and generator voltages. Eight trials were needed in the second iteration when reactive source limits were also enforced.

Twenty three trials were needed in the third iteration when load voltages were included in the enforced inequalities. After the third iteration only five more trials were needed to identify the binding set for the solution. The 131 binding inequality constraints consisted of 14 VAR limits and 117 bus voltage limits.

Iterations seven through twelve were needed only to solve the power flow equations while enforcing a fixed set of binding inequalities. Convergence was slow near the end because of the weakness of the decoupled PF formulation when the PF residuals become small. A change in solution scheme at this point would be advantageous.

It is also interesting to note that the objective function reached its lowest value on the third iteration and then gradually increased as the power flow mismatches were diminished. More detailed monitoring and analysis of what is happening in the course of the solution is expected to provide insight for improving algorithms.

The possibility that OPF solutions may not be unique is of considerable concern. Several OPF problems have been solved with widely different starting conditions and with entirely different algorithms. In all such tests identical solutions were obtained for the same problems. While this does not prove that all OPF problems have unique solutions, it strongly suggests that at least some of them do.

SUMMARY AND CONCLUSIONS

The Newton approach is suitable for development of practical OPF programs. This conclusion is supported by tests of Newton-based algorithms with prototype codes. Although efficiency has not been fully exploited in the prototype codes used in testing, it is possible to estimate the performances of efficient codes for the same operations.

There is always some risk in drawing conclusions about a new approach that has not been extensively tested in production use. It seems unlikely, however, that any difficulties can arise that would invalidate the main outline of the given approach. If an explicit Newton method cannot be made to solve a problem, any existing quasi-Newton method would be no better. The only known alternatives are weaker, more approximate methods that involve more computation.

The key ideas of the given Newton approach are,

1. An explicit Newton formulation.
2. Decoupling.
3. Quadratic penalties for enforcing inequality constraints.
4. Special strategies for finding the binding inequalities.
5. Special sparse matrix/vector techniques.

Some of the attributes of algorithms based on the approach are,

1. Superlinear convergence to the K-T conditions.
2. Solution time comparable to a few conventional PF's.
3. Solution time and matrix storage not adversely affected by number of controls or inequalities.
4. No need for user-supplied tuning factors or interaction to obtain a solution.
5. Adaptability to all three subdivisions of OPF; active/reactive dispatch, active only, reactive only.

A solution algorithm based on the given Newton approach can be roughly divided into,

1. Problem definition: objective function, controls, constraints, etc.
2. Solution formulation: decoupling, approximations, etc.
3. Methods for enforcing inequality constraints, penalties, trial iterations, etc.
4. Strategies for finding the binding inequalities.
5. Sparsity techniques.

In developing an algorithm, choices must be made among the several possibilities in each part. The choices in any one part will be affected by the choices made in others. The number of combinations of choices is large and each application area may require different choices. As more knowledge and experience are gained, the best choices for each application should become evident. Practical, production-grade OPF programs can be developed from what is presently known.

ACKNOWLEDGEMENTS

This research is sponsored by EPRI under contract RP1724-Phase I. ESCA Corporation is the prime contractor. Hermann Dommel, Philip Gill, Walter Murray, Michael Saunders, Larry Stanfel, William Tinney, and Margaret Wright are the consultants to the project. Dr. John Lamont is the EPRI project manager.

Specific appreciation is expressed to Dr. John Lamont for initiating the project and for his continuing guidance; to Walter Murray, Margaret Wright, Philip Gill, and Michael Saunders for their expert advice on the overall application of optimization methods; to Dr. Hermann Dommel for sharing his broad perspective and experience on power system analysis; and to Dr. Larry Stanfel for his investigation of the relationship between the Newton method and Linear Programming in a real time operations environment.

The authors also wish to thank the project's industry advisers for the utility perspective they provided.

APPENDIX

Expressions For Matrix Elements

The matrix W is composed of submatrices J and H. Expressions for J are well known from the Newton PF; expressions for H are less familiar. More than fifty expressions are required for H, but many fewer are required for H′ and H″.

Each element of H′ or H″ is of the form $\partial L/\partial y_k \partial y_m$. Since L is just a combination of PF equations, each term of H′ or H″ is a sum of second partial derivatives of the PF equations wrt, y_k and y_m. For any pair of variables y_k and y_m, nonzero second partial derivatives exist only for those PF equations that contain y_k and y_m. The number of nonzero terms in the sum for any element of H′ and H″ is small. The expressions for representative elements needed for W′ and W″ are developed in the following. Expressions for the first partial derivatives of the PF equations are needed for J′ and J″. Expressions for second partial derivatives of PF evaluations are needed for H′ and H″.

Power Flow Equations

The polar form of the PF equations is shown below.

$$P_k = v_k^2(\hat{G}_{kk} - \Sigma t_{km}^2 G_{km}) + (v_k \Sigma v_m t_{km} Y_{km} \cos\psi_{km})$$

$$Q_k = v_k^2(-\hat{B}_{kk} - \Sigma t_{km}^2 B_{km}) + (v_k \Sigma v_m t_{km} Y_{km} \sin\psi_{km})$$

$(G_{km} + j B_{km})$ Transfer admittance of branch km.

$$Y_{km} = \sqrt{G_{km}^2 + B_{km}^2} \qquad \gamma_{km} = \text{arc tan } B_{km}/G_{km}$$

Summations are over all branches (k,m) connected to bus k.

$\hat{G}_{kk} + j\hat{B}_{kk}$ Driving point admittance at bus k exclusive of contributions from any controllable transformers with their tappers at bus k.

t_{km} If branch (k,m) is a controllable transformer, t_{km} is its tap ratio. Otherwise, $t_{km}=1.0$.

ϕ_{km} If branch (k,m) is a phase shifter, ϕ_{km} is its shift angle. Otherwise, $\phi_{km} = 0$.

In the following derivatives,

$$\psi_{km} = (\theta_k - \theta_m + \phi_{km} - \gamma_{km})$$

First Partial Derivatives of P_k for J′ and g′.

$$\frac{\partial P_k}{\partial \theta_k} = -v_k \Sigma v_m t_{km} Y_{km} \sin\psi_{km}$$

$$\frac{\partial P_k}{\partial \theta_m} = v_k v_m t_{km} Y_{km} \sin\psi_{km}$$

$$\frac{\partial P_k}{\partial \phi_{km}} = -v_k v_m t_{km} Y_{km} \sin\psi_{km}$$

First Partial Derivatives of Q_k for J″ and g″.

$$\frac{\partial Q_k}{\partial v_k} = 2v_k(-B_{km} + \Sigma t_{km}^2 B_{km}) + \Sigma v_m t_{km} Y_{km} \sin\psi_{km}$$

$$\frac{\partial Q_k}{\partial v_m} = v_k t_{km} Y_{km} \sin\psi_{km}$$

$$\frac{\partial Q_k}{\partial t_{km}} = 2v_k^2 t_{km} B_{km} + v_k v_m Y_{km} \sin\psi_{km}$$

Representative Second Partial Derivatives of P_k for H′

$$\frac{\partial^2 P_k}{\partial \theta_k^2} = -v_k \Sigma v_m t_{km} Y_{km} \cos\psi_{km}$$

$$\frac{\partial^2 P_k}{\partial\theta_k \partial\theta_m} = v_k v_m t_{km} Y_{km} \cos\psi_{km} = -\frac{\partial Q_k}{\partial \theta_m}$$

Representative Second Partial Derivatives of Q_k for H″

$$\frac{\partial^2 Q_k}{\partial v_k^2} = 2(-\hat{B}_{kk} + \Sigma t_{km}^2 B_{km})$$

$$\frac{\partial^2 Q_k}{\partial v_k \partial v_m} = t_{km} Y_{km} \sin\psi_{km}$$

Example of Hessian Elements

Following are some of the expressions for elements of H′ and H" for this example problem. Refer to the expression for L of the example in (2).

$$\frac{\partial^2 L}{\partial \theta_2^2} = C_1 \frac{\partial^2 P_1}{\partial \theta_2^2} - \lambda_{P2} \frac{\partial^2 P_2}{\partial \theta_2^2} - \lambda_{Q2} \frac{\partial^2 Q_2}{\partial \theta_2^2} - \lambda_{P4} \frac{\partial^2 P_4}{\partial \theta_2^2} - \lambda_{Q4} \frac{\partial^2 Q_4}{\partial \theta_2^2}$$

$$\frac{\partial^2 L}{\partial v_4 \partial v_5} = -\lambda_{P4} \frac{\partial^2 P_4}{\partial v_4 \partial v_5} - \lambda_{Q4} \frac{\partial^2 Q_4}{\partial v_4 \partial v_5} - \lambda_{P5} \frac{\partial^2 P_5}{\partial v_4 \partial v_5} - \lambda_{Q5} \frac{\partial^2 Q_5}{\partial v_4 \partial v_5}$$

Expressions for other elements of H′ and H" follow directly from these examples.

REFERENCES

1. B. Stott, O. Alsac and J.L. Marinho, "The Optimal Power Flow Problem", in Electric Power Problems; The Mathematical Challenge, Book, SIAM, pp 327-351, Philadelphia, 1980.

2. A. M. Sasson and H. M. Merril, "Some Applications of Optimization Techniques to Power System Problems", IEEE Proceedings, vol. 62, pp. 959-972, July 1974.

3. H. H. Happ, "Optimal Power Dispatch - A Comprehensive Survey", IEEE Transactions on Power Apparatus and Systems, vol. PAS-96, pp. 841-854, May/June 1977.

4. B. Stott, J. L. Marinho and O. Alsac, "Review of Linear Programming Applied to Power System Rescheduling", IEEE PICA Conference Proceedings, Cleveland, pp. 142-154, May 1979.

5. H. W. Dommel and W. F. Tinney, "Optimal Power Flow Solutions". IEEE Transactions on Power Apparatus and Systems, vol. PAS-87, pp. 1866-1876, October 1968.

6. R. C. Burchett, H. H. Happ and K. A. Wirgau, "Large Scale Optimal Power Flow", Ibid, vol. PAS-101, No. 10, pp. 3722-3732, October 1982.

7. A. M. Sasson, F. Viloria and F. Aboytes, "Optimal Load Flow Solution Using the Hessian Matrix", Ibid, vol. PAS-92, pp. 31-41, January/February, 1973.

8. F. L. Alvarado and M. K. Enns, "Blocked Sparse Matrices in Electric Power Systems", Paper A 76 362-4, IEEE PES Summer Meeting, Portland, July 1976.

9. B. Stott and O. Alsac, "Fast Decoupled Load Flows", IEEE Transactions on Power Apparatus and Systems, vol. PAS-93, pp. 859-869, May/June 1974.

10. E. C. Housos and G. D. Irisarri, "A Sparse Variable Metric Optimization Method Applied to the Solution of Power System Problems", Ibid, vol. PAS-101, no. 1, pp. 195-202, January 1982.

11. O. Alsac, B. Stott and W. F. Tinney, "Sparsity-Oriented Compensation Methods for Modified Network Solutions", Ibid, vol. PAS-102, No. 5, pp. 1050-1060, May 1983.

12. P. E. Gill, W. Murray and M. H. Wright, Practical Optimization, Book, Academic Press, 1981.

13. P. E. Gill, W. Murray, M. A. Saunders, M. H. Wright, Sparse Matrix Techniques in Optimization, Report SOL 82-71, Department of Operations Research, Stanford University, California, December 1982.

Biography

Bruce Ashley received his BSEE from the University of Washington in 1980 and attended Purdue University under a Purdue Electric Power Committee Scholarship. He joined ESCA in December, 1981, and is a member of the IEEE PES, IAS, and the Computer Society.

Brian Brewer received his BSEE degree in 1980 and MSEE degree in 1982 from the University of Washington, and is a member of IEEE PES and Tau Beta Pi. He joined ESCA in July 1982.

Art Hughes received BSEE degree from the University of Manitoba in 1972 and MS and PhD degrees from the University of Texas at Arlington in 1976 and 1978. He is presently employed by the ESCA Corp.

David I. Sun received BSEE and MSEE degrees from Resselaer Polytechnic Institute in 1974 and 1976, and PhD from the University of Texas at Arlington in 1980. He is presently employed by the ESCA Corp.

William F. Tinney received BS and MS degrees from Stanford University in 1948 and 1949. He worked for the Bonneville Power Administration from 1950 to 1979 and was head of the System Analysis Section at the time of his retirement. Most of his experience has been in power system computation. He is presently a consultant in this field.

Discussion

S. K. Chang and **J. Lequarre** (Systems Control, Inc., Palo Alto, CA): This paper has presented a refreshing new approach to the solution of optimal power flow problem. At Systems Control, we have developed a Loss Minimization Program (Reactive Control Optimization) employing a similar Newton-based approached and the test results indicate that this method is very promising for practical applications. We would therefore like to share our experience by offering the following comments:

1. Although not stressed in the paper, we feel that a good initial estimate of z is essential for improving the robustness of the algorithm. This is due to the fact that Newton's method is ideal only in terms of its local convergence properties, but not in terms of global convergence. An initial start with flat voltage and zero $\underline{\lambda}$ will generally cause difficult or erratic convergence. A typical scheme would be as follows: Perform one load flow iteration to obtain initial state variables corresponding to a reassigned set of control variables. Then compute the initial $\underline{\lambda}$ from the relation $[J_V]^T\underline{\lambda} = g_V$, where $[J_V]$ is the nonsingular power flow Jacobian matrix, and g_V is the gradient vector of the objective function with respect to the state variables. For the Loss Minimization problem, the initial assignment of maximum voltages to the generator control variables was found experimentally to give best convergence characteristics.
2. The partition in the paper of the W matrix into an external submatrix and a main submatrix suggests the possibility of ill-conditioning if a controllable transformer has negligible conductance. For example, the diagonal term (Δt, Δt) corresponding to a controllable transformer in the Loss Minimization problem can be written as (Δt, Δt)− term $= 2V_k^2\ [g(1-\lambda_{PK}) + b\lambda_{Qk}] = 2V_k^2 \cdot b \cdot \lambda_{Qk} |_{g=o}$ where
   - V_k is the voltage magnitude at the tap-side bus k
   - g,b the transformer conductance and susceptance, respectively
   - $\lambda_{Pk}, \lambda_{Qk}$ the Lagrange multiplier for real and reactive power flow equations at bus k, respectively.

   If bus k is of PV-type and λ_{Qk} is zero, the term (Δt, Δt) is zero and can cause numerical difficulty if the partition strategy is applied. A modified ordering or processing scheme, e.g., placing the tap variable in the main submatrix, is therefore required to circumvent this problem.
3. In spite of its heuristic nature, the strategy of enforcing/releasing multiple constraints at a time seems to be essential for efficient identification of the binding inequality set in large scale OPF problems. To apply this strategy, however, the possible phenomenon of "zigzagging" should be properly guarded against so that a constraint is not repeatedly enforced and released during the trial iterations. Although finding a method for identifying the binding inequalities still is a major challenge, it seems that a strategy that utilizes zigzagging-guarded multiple constraint enforcement/release coupled with deterministic single constraint enforcement or release would be a practical choice. The authors' comments concerning this point will be appreciated.

Finally, wr would like to commend the authors on a well-written and important paper.

Manuscript received February 13, 1984

B. K. Johnson (Power Technologies, Inc., Schenectady, NY): The authors have presented a very interesting paper on methods of optimal power flow solutions. Having developed a technique similar to their coupled solution several years ago,[1] I read the paper with particular interest.

A point to note is that if the objective function, F is total cost of generation then the Lagrange multipliers λ_{pk} amd λ_{pa} equal the incremental cost associated with their particular constraint equation, i.e., the incremental cost of power or vars at system buses where power or vars are constrained. This suggests that this information might be useful for determining the charges which should be assessed for interchanges between utilities or the most economic location for capacitor banks. Have the authors pursued such an extension of their method?

REFERENCE

[1] V. D. Albertson, B. K. Johnson, and W. Scott Meyer, "Exact Economic Dispatch With Quadratic Convergence", IEEE Conference Paper No. 68 CP 671-PWR presented at the IEEE Summer Meeting, June 23-28, 1968.

Manuscript received February 21, 1984

Walter L. Snyder, Jr. (Leeds and Northrup Co., North Wales, PA): The authors are to be complimented on a well developed, practical, and workable approach to the optimal power flow problem. This discussion presents a more compact and generalized mathematical statement of the problem objective, and explores the treatment of active generation for both binding and nonbinding constraints.

The authors' approach minimizes the following Lagrangian type function:

L(z) = L(y,λ) = [C * P] − [λ * dP] + (½) * [dx * [S] * dx] (equation di)

where:

[and] denote row and column vectors, respectively

The d "operator" denotes a finite change, not differentiation

p] = P(y)], and includes both real and reactive power (or either one in a decoupled formulation)

dp] = P(y)] − P(sch)]

dx] = [A] * Y] − x(sch)]

Eq. (di) should be identical to the authors' formulation and lead to the same solution equation, but expresses the problem in matrix form, explicitly including the quadratic penalty functions. The advantage of the form given in (di) is that it allows one to more clearly review and evaluate the complete objective.

It should be pointed out that the lambda terms of (di) contain dP rather than P as in the authors' paper. Could the authors please verify that P in their lambda term is actually a mismatch, dP, while P in their cost term represents the actual injection and not a mismatch. If such is not correct, would the authors please clarify these expressions.

The discusser views C and λ as playing the same role, where both are never non-zero at the same time for a given element of P. If C is zero, λ is a variable which is solved such that P is constrained to a scheduled value by enforcement of the Kuhn-Tucker conditions. A non-zero C, hoever, is viewed as a prespecified, fixed incremental cost thereby taking the place of lambda. For non-zero C, P is no longer constrained to a scheduled value since C is not a variable, rather the C*P term is minimized in conjunction with the rest of the cost terms. For consistency, a scheduled value of 0 could be assumed in the C*P terms, although this scheduled value is never enforced. The approach would tend to reduce the dispatched elements of P to zero were it not for maintenance of a power balance via the J and H sensitivities which relate dispatched P to scheduled P.

The authors' comments on the preceding paragraph would be appreciated. In particular, the treatment of real power C and lambda coefficients with respect to binding and non-binding constraints is not clearly explained in the paper. Are the real power lambdas held at zero while P is dispatched within limits, and are the associated costs set to zero when P is at a limit? If C is left non-zero when P is binding, does the non-zero lambda effectively override the C*P term? In Eq. (4) of the paper, the inclusion of real power but not reactive power lambda terms for buses 1 and 3 lends to the confusion, since both real and reactive power are defined as dispatchable on the two buses.

Similar confusion is evident in Eqs. (6) and (9) of the paper where reactive lambda terms are 'dummied out' (set to zero) for buses 1 and 3 while real power lambda terms are not. Again, this suggests that real power on buses 1 and 3 is scheduled while the paper states that it is dispatchable.

It should be pointed out that the zero diagonal terms in (6) and (9) will present no problem since fill in will occur when the matrix is actually factored due to the jacobian terms of the associated diagonal matrix packet. It is assumed that the only purposes of the arbitrarily large diagonal terms is to ensure a zero correction to lambda and v on execution of the 'divide by diagonal' step of the forward—backward substitution process.

Finally, the last term of Eq. (di) at the beginning of this discussion is seen as a more general expression of the quadratic penalty terms used to enforce limits on the variables themselves as well as on linear functions of the variables. [A] is simply an incidence type matrix (not to be confused with the set, A, of binding constraints), and is a trivial identity for limits on the variables themselves. Was this method considered by the authors for "beyond the scope of this paper" soft enforcement of limits on power sources?

In Eqs. (16a) and (16b) of the paper, should a scheduled power angle term be included as was done in Eq. (15)?

Overall, the presentation of this paper was still excellent and was extremely motivating to the discusser, injecting new life into the optimal power flow problem.

Manuscript received February 23, 1984

B. Stott, O. Alsac and **A. Bose** (Electrical & Computer Engineering, Arizona State Univ., Tempe, AZ): For over twenty years, researchers have had limited success in developing efficient, reliable solution methods for large "classical" OPF problems with nonseparable objective functions. These problems are encountered mainly when scheduling reactive power for the minimization of objectives such as production cost or losses.

The present paper introduces an approach with real potential for a breakthrough. In doing so it reverses a recent trend towards the use of sophisticated modern general-purpose constrained-optimization methods and/or codes, which appear to solve OPF problems correctly but relatively very slowly. High speed is aimed for by reverting to optimization first principles and focussing the development effort on structure-suited implementation, resulting in an approach with good convergence power and excellent network sparsity preservation.

The attraction of the approach centers on the facility with which the equality-constrained OPF problem can be solved. The Newton solution of its Lagrangian necessary-condition equations is conceptually simple, performs second order minimization with no tuning, and is amenable to efficient exploitation of problem structure. Unfortunately, this approach has no accompanying powerful well-established means for handling inequalities. Therefore heuristic schemes, tailored to particular OPF formulations, need to be developed for switching limit constraints into and out of the binding set via equation addition/removal or quadratic penalties.

The same general approach has been tried in small-scale research a number of times before (see especially [A]). Among the major improvements introduced by the present work is a superior way of organizing the necessary-condition equations for efficient decoupled sparse solution. It is shown that quadratic penalties can be used effectively for state-variable limits, as well as for functional inequalities. Improvements in efficiency through the use of advanced sparse matrix techniques are emphasized.

The most critical factor for success in the approach is to find schemes that determine the binding limit set reliably and rapidly, on a wide range of power systems, constraint types and OPF formulations. This is precisely the major problem encountered in many previous OPF methods and, as far as is known, never yet satisfactorily resolved. The present paper emphasizes this as the key problem that must be overcome by further development. It gives very rough outlines of several candidate trial-iteration schemes.

As the paper itself states, it is introducing a methodology as much as a developed method. Many questions of detail remain to be answered. A few of these questions are as follows. It is stated that enhancement of positive-definiteness of the Lagrangian is unnecessary, implying the absence of convergence problems. Has sufficient experience with different systems and situations been accumulated to verify this? Is decoupling sensitive to branch R/X ratios? What is the effect of decoupling when voltage magnitude limits need to be enforced by active power scheduling. What is the infeasibility behavior of the approach, and what infeasibility strategies are most appropriate with it? Penalties are advocated for functional inequalities, but other than for simple line limits, they can alter the structure of matrices W' and W''. How are important limits such as interchange, reserve and contingency constraints to be handled? How is solution oscillation avoided using piecewise cost functions as in Eq. (1)? Can controlled hvdc links, with variable converter-transformer taps, conveniently be included? Can the approach conveniently and efficiently accommodate all modeling and local-control features of a conventional power-flow solution, when needed?

In conclusion, now that the approach's potential has been clearly exposed, its general effectiveness and applicability for industrial use need to be established. There are many practical requirements for OPF solutions not covered by the basic classical problem formulation. Extensive further development and testing with different power systems, objectives, constraints, etc., will be necessary. We look forward to further results from the project.

REFERENCE

[A] H. Duran, "A Simplex-like method for solving the optimum power flow problem," *Proc 8th PICA Conf.*, pp. 162-167, Minneapolis 1973.

Manuscript received February 24, 1984

W. W. Lemmon and **W. R. Barcelo** (Middle South Services, Inc., New Orleans, LA): The authors' contribution to the field of Newtonian OPF methods is appreciated. As the authors point out, many Newtonian approaches are possible, but a simplification of the true bordered Hessian matrix is essential since it has 12 non-zero entries for each line between (P, Q) type buses. In 1977 [1] we presented a Newtonian-based OPF method for the real-time enforcement of security constraints using the bordered Hessian matrix. Our method uses the highly sparse "diflex" approximation which assumes that the (P,Q) busses have passive loads. In this method, a compensation term is added to the Hessian to account for voltage deviations during solution iterations. This approach required only 4 matrix entries per transmission line instead of the 12 entries required by a pure Newtonian method. We have used the method in a study mode to optimally dispatch active power while maintaining security constraints. The logic is available for both active and reactive dispatching [2] but the reactive part has not yet been implemented. Our constraint enforcing technique is basically "Algorithm II" for active source limits and security constraints and "Algorithm III" (modified) for reactive sources and voltage constraints. Algorithm II is significant in that a proof of convergence exists.[1]. The Objective Function is production cost and refactorization is both selective and compensational. P-Q decoupling is performed only on an appropriate subset of busses; also, the $P\Theta$ and QV parts are subjected to simultaneous "trial iterations" (which we called "subiterations").

The authors' use of relaxation on the $P\theta$ and QV subsystems is analogous to the decoupled PF technique and in our opinion has high potential. However, when security constraints (special function constraints) are being held, the decoupling assumption can fail for the Hessian submatrix H. This "recoupling" is independent of whether the constraints are held by λ's or by penalty functions. In the presence of recoupling, limiting trial iterations to individual $P\theta$ or QV cycles, as the authors propose, may lead to inappropriate selection of the binding inequality constraints due to each subsystem's lack of knowledge of the other's constraint enforcement capabilities. This may result in an unduly large number of trial iterations or even divergence. We shall describe the reason for this recoupling phenomenon and suggest three possible solutions.

Recoupling occurs when a constraint is affected strongly by both voltage and angles—we call this *intrinsic* recoupling. For example, when trying to impose a thermal constraint such as MVA or current magnitude on a line that is transmitting appreciable reactive power, a strong (θ, V) coupling term is added to the Hessian. The authors' example of constraining real power flow does not show this effect.

Recoupling also can occur whenever constraints are being held tightly enough to cause the resultant λp profile to become distorted. We call this λ - *differential recoupling.* In order for θ, V) terms in the Hessian to be negligible, it is necessary that the (λp difference between connected busses be reasonably small—which is assured in the absence of security constraints by low line losses. While holding certain security constraints, we have observed λp's between connected busses that are orders of magnitude apart! This type of recoupling occurs because frequently voltages are helpful in controlling constraints that depend explicitly only on angles: for example, raising voltages (within allowable limits) has the ultimate effect of decreasing angular differences.

Because of the great potential of the decoupling approach, we have attempted to develop solutions to the recoupling problem. Three of the more promising solutions which may be applicable to the authors' technique are as follows:

Solution 1: Give the reactive subsystem "busy work". If all busses have sufficiently large non-relaxable quadratic penalties on their voltages, the coupling terms are made negligible because voltage can no longer be used to help maintain security constraints. Unfortunately, this solution uses reactive power only for voltage profile control, which violates the intent of the OPF solution.

Solution 2: Partial decoupling without relaxation. This is the solution which we have provisionally chosen for our optimizer. The bordered Hessian W is decoupled except for the recoupling terms and iterations are performed simultaneously on the $P\theta$ and QV subsystems. Of course, when no recoupling is present, it would be desirable to relax on the two subsystems in the normal manner since simultaneous iteration takes about twice as long for given accuracy criteria.

Solution 3: Multiple block relaxation. Normally, the recoupling terms (if any) are not very numerous. Thus it should be possible to decompose the bordered Hessian, into, for example, three approximately decoupled submatrices W', W'', and W''', where W''' spans a relatively small set of $P\theta$ and QV variables that are strongly coupled. Each iteration would then comprise a relaxation on three subsystems: a large $P\theta$, QV pair and a small set of connected mixed variables.

We would appreciate any comments the authors may have concerning the recoupling problem and our proposed solutions. Also, would the authors please comment on the accuracy of the matrix factorization in their approach when large weighting factors are required on the penalty functions to hold the inequality constraints and how these weighting factors are selected.

We commend the authors for their organization of the material on the constrained OPF problem and their presentation of the variety of algorithms possible for determining the binding inequality set. We look forward to further papers on this subject by the authors and are especially interested in the convergence properties of the decoupled approach.

REFERENCES

1. W. R. Barcelo, W. W. Lemmon, and H. R. Koen, "Optimization of the Real-Time Dispatch with Constraints for Secure Operation of Bulk Power Systems." *IEEE Trans. Power App. Syst.*, vol. PAS-96, No. 3, p. 741-757, May/June 1977.
2. W. R. Barcelo, W. W. Lemmon, and J. D. Stenquist, *The Middle South Tulane Security Model for a Bulk Power Management System*, Middle South Services Proprietary Documentation, 1975.

Manuscript received February 25, 1984

A. Monticelli and **F. F. Wu** (Dept. of EECS, Univ. of California): The authors have developed a fast, accurate, and robust approach to optimal power flow. The method has potential for many other applications. For example, it can be applied to the solution of state estimation problem with equality constraints [A], where at each iteration a matrix equation of the form of Eq. (7) is solved.

The Newton method takes a large step at each iteration. In the OPF by Newton's approach developed in this paper, the direction of movement at each iteration depends on the selection of binding constraints. If one takes the Newton's direction based on the binding constraints at the current solution point, due to constraint violation, the stepsize may be very small (Algorithm II). The trial iteration is a scheme to determine which Newton's direction is better. It seems to us the scheme used in the trial iteration is crucial to the success of the method. We would appreciate if the authors would comment on: (1) the comparison between Newton-OPF/Algorithm II with the conventional gradient approach, (ii) to elaborate on the method they used for trial iteration, and (iii) to comment on whether the gradient approach can benefit from the use of trial iteration.

REFERENCE

[A] F. A. Aschmoneit, N. M. Peterson, and E. C. Adraian, "State Estimation with Equality Constraints," *Proc. 1977, PICA Conference*, Cleveland, OH, pp. 427-430.

Manuscript received March 2, 1984

W. F. Tinney, D. I. Sun, B. Brewer, and **A. Hughes:** We are grateful for the excellent discussions. This closure first addresses issues common to several of the discussions, followed by replies to other questions listed under the discussers' names.

We apologize for omissions in the bibliography, as some papers describing similar approaches have gone unreferenced. We've found that it is difficult to limit such a list once it is started. Some of the unreferenced papers are listed in the discussions, but more exist.

In retrospect, it is apparent that several forms of the Newton approach have been developed previously by others; our contribution lies in developing the techniques for making it practical. We had an advantage over other investigators in having an EPRI research contract that enabled us to carry out the costly programming and testing.

The importance of techniques cannot be overemphasized. The method and techniques are inseparable, but the equations and matrices given in the paper do not explain the techniques. For highest efficiency, sparsity techniques must be adapted to each specific application; general purpose software cannot do this.

S. K. Chang and *J. Lequarre:* We are pleased that the discussers have developed a similar Newton approach, and that their experience with it confirms our claims about its promise.

The suggested idea of performing one conventional power flow iteration with nominal control settings before optimizing seems to work well as a starting procedure. A converged power flow solution or a feasible state is unnecessary.

As pointed out, a controllable transformer with negligible resistance will introduce a near zero diagonal term that cannot be processed in the given matrix partitioning without numerical breakdown. Putting such transformers in the main submatrix corrects the numerical difficulty in most cases, but may degrade efficiency. If node k in the discussers' equation is radial as well as PV-type, a local degeneracy will occur with a no-resistance transformer even if it is in the main submatrix. All transformers have sufficient resistance to prevent numerical breakdowns. If the resistance is missing in the data, a minimum value can be automatically defaulted. Resistance should always be modeled accurately for loss minimization. The recommended partitioning is highly advantageous, and it can be retained for all transformers.

We concur with the cautions about zigzagging in strategies that enforce/release multiple constraints simultaneously, although if a good active set management strategy is used, zigzagging should occur only rarely. When it does occur, a simple ad hoc scheme would be to keep count of the individual zigzags and to reduce the number of constraints being simultaneously changed should a specified count limit be exceeded. In a particularly difficult situation, this scheme could eventually reduce to changing one constraint at a time for the given iteration, although this, also, should rarely occur.

B. K. Johnson: The cited paper co-authored by the discusser was probably the earliest application of the Newton approach to the OPF problems.

The Lagrange multipliers of the power flow equations could, as suggested, be used in various ways as a basis for transaction-costing which takes into account incremental transmission as well as production costs. The OPF is also a key module in important applications that have been impractical until now because of the lack of an efficient OPF solution; capacitor allocation is one such application. The Newton approach should be highly efficient in this OPF subproblem.

W. I. Snyder, Jr.: A more detailed exposition of the basic formulation than appears in the paper is given here in an attempt to answer the questions raised in the discussion. A new variable for dispatchable active power is introduced and a quadratic cost function is used instead of piecewise linear costs.

The following definitions of symbols are for a bus with dispatchable active power:

PG_k	Active power generation (variable)
P_k	Active power mismatch (function) $= PG_k - \Sigma P_{km}$
f_k	Cost of generation (function) $= \frac{1}{2} a_k (PG_k)^2 + b_k(PG_k)$
C_k	Incremental cost (function) $= a_k(PG_k) + b_k$
F	Objective function $= \Sigma f_k$

Using these definitions the Lagrangian for the five bus example becomes,

$$L = f_1 + f_3 - \lambda_{P_1}P_1 - \lambda_{P_2}P_2 - \lambda_{P_3}P_3 - \lambda_{P_4}P_4 - \lambda_{P_5}P_5 - \lambda_{q_2}Q_2 - \lambda_{q_4}Q_4 - \lambda_{q_5}Q_5 \quad \text{(C.1)}$$

In (C.1) the variable PG_1 occurs in functions f_1 and P_1, and the variable PG_3 occurs in functions f_3 and P_3. Assume that axes for the variables

PG_1 and PG_3 are added to matrix W′ for the decoupled version. The conditions for bus 3 within W′ are shown in (C.2).

$$\begin{bmatrix} \ddots & & & \\ & \partial^2\mathcal{L}/\partial PG_3^2 & & -1 \\ & & h & j \\ & -1 & j & 0 \\ & & & & \ddots \end{bmatrix} \begin{bmatrix} \vdots \\ \Delta PG_3 \\ \Delta\theta_3 \\ \Delta\lambda p_3 \\ \vdots \end{bmatrix} = \begin{bmatrix} \vdots \\ -\partial\mathcal{L}/\partial PG_3 \\ -\partial\mathcal{L}/\partial\theta_3 \\ -\partial\mathcal{L}/\partial\lambda p_3 \\ \vdots \end{bmatrix} \qquad (C.2)$$

where $$\frac{\partial\mathcal{L}}{\partial PG_3} = \frac{\partial F}{\partial PG_3} - \lambda p_3 = C_3 - \lambda p_3$$

and $$\frac{\partial^2\mathcal{L}}{\partial PG_3^2} = \frac{\partial^2 F}{\partial PG_3^2} = a_3$$

The axes for dispatchable active power, PG_1 and PG_3, do not appear explicitly. They are reduced into the rest of the matrix. Eliminating the axis for PG_3 gives,

$$\begin{bmatrix} \ddots & & \\ & h & j \\ & j & -\left(\partial^2\mathcal{L}/\partial PG_3^2\right)^{-1} \\ & & & \ddots \end{bmatrix} \begin{bmatrix} \vdots \\ \Delta\theta_3 \\ \Delta\lambda p_3 \\ \vdots \end{bmatrix} = \begin{bmatrix} \vdots \\ -\partial\mathcal{L}/\partial\theta_3 \\ -\partial\mathcal{L}/\partial\lambda p_3 - \left(\partial^2\mathcal{L}/\partial PG_3^2\right)^{-1}\left(\partial\mathcal{L}/\partial PG_3\right) \\ \vdots \end{bmatrix}$$

To initialize the solution, the incremental costs, C_1 and C_3, are evaluated for the initial values of PG_1 and PG_3, and λp_1 and λp_3 are set to the values of C_1 and C_3. Note that for linear generation cost model (as in the example of the paper), the term $\left(-\partial^2 F/\partial PG_k^2\right)^{-1}$ is replaced by a large number, ∞, on the diagonal and the RHS of λp_k axis. The result of these steps is symbolized in Eq. (11a) of the paper. Their effect is to force a predetermined change in $\Delta\lambda p_1$ and $\Delta\lambda p_3$ in the solution, in the same manner as explained for penalties on variables. Solution of the matrix equation ensures coordinated changes in all of the variables.

On each iteration, PG_1, PG_3, λp_1 and λp_3 are updated, and C_1 and C_3 are re-evaluated using the updated values of PG_1 and PG_3. At the K-T conditions $\lambda p_1 = C_1$ and $\lambda p_3 = C_3$. If a bus with dispatchable active power reaches a limit, it is converted to a P-scheduled bus.

Since there is no cost for dispatchable reactive power in the example, the λp_k axes of these buses are simply dummied out. If a reactive source reaches a limit, it is converted to a Q-scheduled bus (the λq_k axis is re-inserted).

The interpretation of the formulation given in the discussion is also essentially correct for a Newton OPF approach and it may be viewed as PF extensions. The seemingly more complicated formulation given here is used mainly in order to express the functions of dispatchable power costs in terms of the variables PG_k instead of the functions P_k. IF P_k were used instead of PG_k in f_k, it would create many additional terms in the matrix. The formulation also has important advantages for some special OPF applications.

As pointed out, the zero diagonals in each diagonal block will fill in before they become pivots in the factorization. However, as an additional safeguard against numerical problems in the test program each block submatrix was processed as a unit, and the diagonal blocks were explicitly inverted. Processing block submatrices is somewhat less efficient than processing individual elements. It is probably an unnecessary precaution, but we are not certain.

The schedules angle terms were omitted in Eq. (16). The equation should be,

$$\frac{\partial\alpha_i}{\partial\theta_k} = s_i(\theta_k - \theta_m - \bar{\theta}_{km}) \qquad (16a)$$

$$\frac{\partial\alpha_i}{\partial\theta_m} = -s_i(\theta_k - \theta_m - \bar{\theta}_{km}) \qquad (16b)$$

These discussions were very helpful in pointing out ambiguities in the problem formulation. We hope this response has answered the questions satisfactorily.

B. Stott, O. Alsac and A. Bose: This discussion summarizes the advantages of the approach and raises important questions about it. The quest for improvement in schemes for determining the binding constraints could continue indefinitely, but the existing ad hoc schemes are adequate for practical purposes. The lack of a definite scheme for automatic adjustment of controls in conventional power flow programs has not prevented their practical use. Finding the binding constraints is a challenge in all optimization methods. It is not unique to the OPF or the Newton approach.

Although the Newton formulation seems to be strongly positive, definite safeguards to ensure positive definiteness should be provided. Augmenting the Lagrangian with additional quadratic penalties is a possible scheme.

Decoupling in the Newton OPF, similar to that in the fast decoupled power flow, is degraded by high R/X ratios. The example problem in the paper had many lines with high R/X ratios. Reducing the ratio in one critical line reduced the number of main iterations for the example solution from 12 to 8. Presumably, the remedies for high ratios used in the decoupled power flow will also work in the OPF.

After the first few main iterations in which the binding constraints are approximately determined, the bordered Hessian submatrices W′ and W″ become nearly constant and no longer have to be recomputed and refactorized. If changes are necessary they can be made by highly efficient partial refactorizations. This considerably reduces the cost of additional iterations needed to reduce the power flow mismatches resulting from high R/X ratios. It also makes solutions of all problems faster.

Some complicated functional constraints, such as area interchange and MW reserve, are not necessarily best enforced by explicitly including quadratic penalties in the bordered hessian. The complications arise from significantly disturbing the sparsity structures of the matrices, and/or from numerical considerations. The alternatives include applying the Schur complement technique, or explicitly introducing additional equations to the matrices. Schur complement is attractive when the number of equations is small; embedding the equations directly is efficient, but the programming is further complicated by the nonuniform block structures.

Various OPF applications will require all of the models now used in conventional power flow programs except certain local controls that would not make sense in a global optimization. It seems reasonable that all models for the Newton power flow could also be implemented in the Newton OPF but they have not been individually evaluated. Multiterminal hvdc links could be difficult. Another obstacle is how to optimize sensibly in a network with equivalents.

Piecewise cost functions appeared in the paper only for illustration; quadratic costs were used in testing. If costs are supplied in piecewise form, it is best to automatically fit them with analytic functions from which first and second derivatives can be computed. Handling of explicit piecewise linear costs the solution process requires needs logic to prevent oscillations at their breakpoints.

The very similar approach to OPF by Duran was overlooked. With good sparsity techniques it could have worked much the same as our approach. He apparently tested it only on small problems.

W. W. Lemmon and W. R. Barcelo: The discussers' successful method for active power dispatch by the Newton-based diflex method was helpful to us and it should have been listed as a reference. We still do not fully understand the diflex approach, and in our investigation we could not see how it could be adapted to enforce voltage constraints efficiently. A comparison between the diflex OPF and our decoupled version for the active/reactive problem should be interesting. The diflex OPF for active power seems to be the first Newton approach used on large practical problems.

Although the exact Newton formulation without decoupling results in a large bordered Hessian matrix, it is not intractable even for the largest power networks. Whether to use the exact formulation or some decoupled approximation would be decided on the basis of computational tradeoffs. Decoupled formulations usually are more efficient, but the exact formulation need not be ruled out.

The need for "recoupling" could be a serious impediment to the success of decoupled approximations for some applications. We greatly appreciate the discussers' sharing of their experience with it and their suggestions for solving it. Either of their suggested solutions 2 or 3 should accomplish recoupling with a minimum sacrifice of the advantages of decoupling. Efficient programming is the main challenge. The fact that the constraints requiring recoupling are generally few in number is a big help. We are inclined towards Solution 3 because we can visualize an efficient way to program it. When recoupling is not necessary, the program would be unaffected.

To enforce simple-bound limits on a non-degenerate set of variables, quadratic penalties are effective. Numerical accuracy is unaffected by the number of binding limits or by larger than necessary penalty weighting factors. It is best, however, to have the weighting factor no larger than necessary since this makes efficient removal of a penalty by compensation (or factor update) a more accurate process. A value of 1,000 times the Hessian diagonal may be a good compromise. The exact value is not critical.

In enforcing functional limits with quadratic penalties, which creates a large off-diagonal as well as diagonal terms, it is possible to experience numerical difficulties. If this occurs, penalty weightings could be reduced in magnitude and shifted to produce the desired degree of enforcement. For this reason, penalties do not have to be extremely large to be effective. Further experience will contribute to the knowledge about how penalties can and cannot be used.

A. Monticelli and F. F. Wu: Some non-OPF problems have matrix formulations identical to those of the bordered Hessian but they are not solved in the manner described in the paper. State estimation is one example. The suggestion that there might be advantages in solving these problems in the same factorized block form as the OPF seems well worth investigating. This form of solution would remove most restrictions on the form of the upper left hand submatrix.

A definitive strategy for trial iterations has not been worked out. Performance of a strategy depends on the number of kinds of constraints enforced/released and the number and kind of quantities monitored on each trial. Programming tecniques are as important as the selection strategy and each affects the other. The strategy resulting in the fewest trials is not necessarily the best since the speed of a trial must also be considered.

The weakness of reduced gradient methods for OPF is such that they tend to hang up before reaching the exact K-T conditions. The hang-up point is not unique and it depends on starting conditions. The difficulty seems to be that the reduced gradient is not in the direction of the minimum; i.e., it does not provide a good move direction. Trial iterations would not help because a gradient-based move constantly encounters constraints that do not lie between the current point and the minimum

Manuscript received April 30, 1984