

Power Flow Solution by Newton's Method

WILLIAM. F. TINNEY, SENIOR MEMBER, IEEE, AND CLIFFORD E. HART, MEMBER, IEEE

Abstract—The ac power flow problem can be solved efficiently by Newton's method. Only five iterations, each equivalent to about seven of the widely used Gauss-Seidel method, are required for an exact solution. Problem dependent memory and time requirements vary approximately in direct proportion to problem size. Problems of 500 to 1000 nodes can be solved on computers with 32K core memory. The method, introduced in 1961, has been made practical by optimally ordered Gaussian elimination and special programming techniques. Equations, programming details, and examples of solutions of large problems are given.

INTRODUCTION

THE AC power flow problem can be solved by several different methods.^{[1-3], [5], [7], [8]} In a recent survey^[9] these methods were classified as either direct or iterative. Actually, all the methods are iterative in one sense because the basic problem involves the solution of a system of nonlinear equations. However, the so-called direct methods employ the direct solution of a related linear system in the iterative algorithm, whereas the iterative methods use a scheme of successive displacements, generally known as Gauss-Seidel. As the survey pointed out, equitable comparisons between methods are difficult because of differences in computers, programming methods, and test problems. In general, the direct methods converge in few iterations and are not subject to ill-conditioned situations. On the other hand, their memory requirements and computing time increase as some power of problem size, thereby limiting their effectiveness to small problems.

The iterative methods converge slowly and are subject to ill-conditioned situations. Their memory requirements are minimal and directly proportional to problem size, but the number of iterations for solution increases rapidly with problem size. However, for large problems only the iterative methods have proved practical. Now that larger systems than ever before are being studied, the need for a better method is becoming increasingly urgent. The purpose of this paper is to describe an improved version of one of the previously published direct methods which offers a definite margin of advantage over other methods for any size or kind of problem. It has been used at Bonneville Power Administration (BPA) since 1962.

In 1961 Van Ness and Griffin described what they called the elimination method of power flow solution.^[7] Actually, their method is the n -dimensional analog of Newton's method^{[14], [17]} and it seems best to refer to it by this name. Using relatively small problems, they showed that this method overcomes deficiencies in the successive displacements methods and has other favorable characteristics. They anticipated that the method would prove equally effective on large problems. However, since the iterative algorithm involved the direct solution of a system of sparse simultaneous equations, the relationship between problem size and computer time and memory requirements could only be established by experiment.

Shortly after its publication, the method was tested at BPA. A hastily written subroutine employing the method was incorporated in the BPA power flow program in place of a suc-

cessive displacements method and tests were run using various sizes and kinds of problems. It was found that the computer time and memory requirements increased rapidly with problem size. The practical upper limit for the method on computers with 32K core memory was found to be about 200 nodes and the time required for solution of problems of this size was greater than with the best accelerated successive displacements methods. However, it was also found that the method could solve problems which, for unknown reasons, the successive displacements methods could not solve.

It was evident that the main difficulty was not in the method itself but in the elimination procedure for solving the simultaneous equations. Recognition of this fact led to the discovery of a simple but effective remedy which was reported in 1963.^[10] Briefly, it is a scheme for arranging the order of the rows of a matrix prior to triangularization by Gaussian elimination such that the accumulation of new terms is kept to a near minimum. This idea is of great importance in the solution of all large power system network problems and several other papers have been written on it.^{[11]–[13]} Application of this scheme, which will be referred to as optimally ordered elimination, greatly improved the efficiency of Newton's method for solution of the power flow problem and was immediately adopted for use at BPA. Subsequently the program has been rewritten several times and many improvements have been added.

CHARACTERISTICS OF METHOD

The characteristics described in this section will be discussed in more detail in sections that deal with aspects of the method that affect them.

Speed

The number of iterations required for solution is virtually independent of problem size and kind. This is strictly true only for problems with a flat voltage start and without automatic adjustments. Under these conditions an acceptable solution can be obtained in four or five iterations. With optimum machine language programming one iteration of this method is equal to about seven of the successive displacements methods. With a good starting approximation, solutions can be obtained in fewer iterations. When extensive program-controlled adjustments such as transformer taps, area interchange, and constraint checks are required, solutions may take up to three times as many iterations. The time per iteration for a 500-node problem on the IBM 7040 is about 11 seconds. The high speed of solution is due to the quadratic convergence characteristic of Newton's method, optimally ordered elimination, and special programming techniques.

Accuracy

The accuracy of the power flow solution is limited only by the round-off error of the direct solution of the system of simultaneous equations. Using single precision 36-bit floating point words, the solution of a 500-node problem will converge to a total absolute mismatch of less than 0.0001 per unit complex power. The concept of mismatch is irrelevant to this method.

Computer Requirements

Although Newton's method takes more memory than the iterative methods, the requirement is not prohibitive. With optimally ordered elimination, the problem dependent memory and time requirements for large systems vary approximately in

Paper 31 TP 67-46, recommended and approved by the Power System Engineering Committee of the IEEE Power Group for presentation at the IEEE Winter Power Meeting, New York, N. Y., January 29–February 3, 1967. Manuscript submitted October 27, 1966; made available for printing June 26, 1967.

The authors are with the U. S. Department of the Interior, Bonneville Power Administration, Portland, Ore.

direct proportion to problem size. This has been found true for systems of up to 1000 nodes and there is reason to expect that it holds for systems of greater size. With the programming scheme used at BPA, a 500-node problem needs up to about 8000 more words of high-speed storage than would be required by the successive displacements methods; twice this amount should be sufficient for a 1000-node problem. Because of the few iterations involved, the program is well-adapted to extensive use of tape for storage of data needed for adjustments. The BPA program for the IBM 7040 with 32K core memory is dimensioned conservatively for 500 nodes (exclusive of passive nodes with only two branches which are eliminated on input and restored on output). Its capacity could be increased to 700 nodes without changing the basic scheme. A less versatile 1000-node program has been written, but this requires data packing and more use of tape storage. Other schemes are possible which could extend the method to about 2000 nodes on a 32K computer; they would be less efficient, but still competitive.

Ill-Conditioned Problems

One of the main deficiencies of the iterative methods is the inability to solve problems with negative transfer reactances which cannot be netted out with the positive reactance of a line. Such reactances arise in the equivalents of 3-winding transformers. Also, the iterative methods are affected adversely by problems in which high and low impedance branches terminate on the same node. Newton's method is unaffected by these conditions and experience at BPA has revealed no other ill-conditioned situations.

Extensions of Method to Other Problems

Probably the most significant attribute of Newton's method is the fact that the program needs only small modification to perform other important functions. For example, the marginal costs of the production of real and reactive power or voltage at each node and the marginal costs of adjustable parameters, such as transformer taps, may be computed quite easily with only small program additions. The incremental sensitivity of any quantity, such as a load voltage or branch power flow, to the variation of any other adjustable parameter, such as a transformer tap or generator voltage, may be computed with more extensive additions. With the availability of these factors it is possible to obtain an optimally adjusted power flow solution. This has been accomplished in test programs and will be reported in another paper.^[20]

Example Solution

Table I shows the iterations for a typical unadjusted solution of a problem of 487 nodes with a flat voltage start. The sums of the absolute real and reactive power errors prior to each iteration are given for five iterations. The error at the end of the fifth iteration is not computed or shown, but the solution is known to

TABLE I

SUMMARY OF ITERATIONS FOR UNADJUSTED 487-NODE PROBLEM*

Iteration Number	Sums of Absolute Residuals Before Each Iteration		Nodes with Error < 0.001 o/1
	P Error o/1	Q Error o/1	
1	375.351 80	372.218 06	480
2	63.436 66	126.186 76	466
3	10.436 66	14.039 96	275
4	0.614 67	0.566 97	52
5	0.002 51	0.002 27	0

* Problem of 487 nodes and 711 branches. Total time for five iterations—51.57 seconds on IBM 7040. No adjustments. One per unit power equals 100 MVA.

be exact except for round-off. In this and the other examples the problem is considered to be solved when the P and Q error at every node is less than 0.001 per unit. It should be noted that the problem is actually solved at the end of the fourth iteration, but this condition is not detected by the program until the end of the fifth iteration. It should also be noted that the rate of convergence increases with each iteration. There is no way to compare this solution with that of the methods of successive displacements because the problem cannot be solved by these methods. The time of 51.57 seconds includes only the five iterations of the solution and not the input-output.

BASIC METHOD

The notation used here follows that of Van Ness and Griffin.

Equations to be Solved

The relationship between node current \bar{I}_k and node-to-datum voltage \bar{E}_k in a network of N nodes is given by the linear equation

$$\bar{I}_k = \sum_{m=1}^N \bar{Y}_{km} \bar{E}_m \quad (1)$$

where \bar{Y}_{km} is an element of the admittance matrix and the super-bar indicates complex quantities.

Complex power at node k is given by

$$(P_k + jQ_k) = \bar{E}_k \sum_{m=1}^N \bar{Y}_{km} \bar{E}_m^* \quad (2)$$

where P_k and Q_k are, respectively, the real and reactive power entering node k , * means complex conjugate, and $j = \sqrt{-1}$.

In the power flow problem it is necessary to solve a system of $(N - 1)$ equations of the form of (2) with certain additional conditions. Since the equations are nonlinear, an iterative scheme must be employed. Newton's method is applicable to certain systems of nonlinear equations if the corresponding Jacobian matrix^[16] can be evaluated and a sufficiently good starting approximation is possible. Both these conditions are fulfilled in the power flow problem.

In the remaining presentation the 7-node network of Fig. 1 will be used as a specific example.

To solve the power flow problem for the example, it is necessary to solve a set of six equations of the form of (2) for the unknown voltages. From the voltages and the input data all other desired output quantities may be calculated directly.

Jacobian Matrix

Newton's method involves repeated direct solutions of a system of linear equations derived from (2). The Jacobian matrix of (2) gives the linearized relationship between small changes in voltage angle and magnitude, $\Delta\delta_k$ and $\Delta E_k/E_k$, and small changes in real and reactive power, ΔP_k and ΔQ_k . For the sample network the matrix equation is as shown in (3).

$$\begin{bmatrix} \Delta P_2 \\ \Delta Q_2 \\ \Delta P_3 \\ \Delta P_4 \\ \Delta Q_4 \\ \Delta P_5 \\ \Delta Q_5 \\ \Delta P_6 \\ \Delta P_7 \\ \Delta Q_7 \end{bmatrix} = \begin{bmatrix} H_{22}N_{22} & H_{23} & & H_{25}N_{25} & & \\ J_{22}L_{22} & J_{23} & & J_{25}L_{25} & & \\ H_{32}N_{32} & H_{33} & H_{34}N_{34} & & H_{36} & H_{37}N_{37} \\ & H_{43} & H_{44}N_{44} & H_{45}N_{45} & & H_{47}N_{47} \\ & J_{43} & J_{44}L_{44} & J_{45}L_{45} & & J_{47}L_{47} \\ H_{52}N_{52} & & H_{54}N_{54} & H_{55}N_{55} & H_{56} & \\ J_{52}L_{52} & & J_{54}L_{54} & J_{55}L_{55} & J_{56} & \\ & H_{63} & & H_{65}N_{65} & H_{66} & \\ & H_{73} & H_{74}N_{74} & & & H_{77}N_{77} \\ & J_{73} & J_{74}L_{74} & & & J_{77}L_{77} \end{bmatrix} \begin{bmatrix} \Delta\delta_2 \\ \Delta E_2/E_2 \\ \Delta\delta_3 \\ \Delta\delta_4 \\ \Delta E_4/E_4 \\ \Delta\delta_5 \\ \Delta E_5/E_5 \\ \Delta\delta_6 \\ \Delta\delta_7 \\ \Delta E_7/E_7 \end{bmatrix} \quad (3)$$

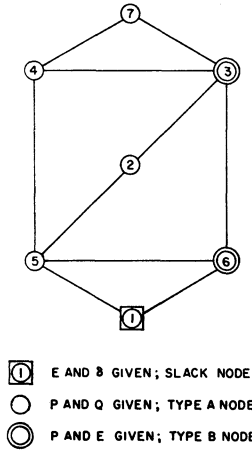


Fig. 1. Sample 7-node network.

The matrix is partitioned in a manner that corresponds with the way it is handled in the program.

The elements of (3) are defined in (4).

$$\begin{aligned} H_{km} &= \frac{\partial P_k}{\partial \delta_m} \\ N_{km} &= \frac{\partial P_k E_m}{\partial E_m} \\ J_{km} &= \frac{\partial Q_k}{\partial \delta_m} \\ L_{km} &= \frac{\partial Q_k E_m}{\partial E_m} \end{aligned} \quad (4)$$

The partial derivatives defined in (4) are real functions of the admittance matrix and the node voltages. Although based on a polar formulation of the problem, they should be computed by rectangular complex arithmetic. There is an alternative rectangular form of the method that has not been used because it requires more memory. Rectangular expressions for the admittance, voltage, and current are

$$\begin{aligned} \bar{Y}_{km} &= (G_{km} + jB_{km}) \\ \bar{E}_m &= (e_m + jf_m) \\ \bar{I}_m &= (a_m + jb_m) \end{aligned} \quad (5)$$

Terms of (4) where m is not equal to k are

$$\begin{aligned} (a_m + jb_m) &= (e_m + jf_m)(G_{km} + jB_{km}) \\ H_{km} &= L_{km} = a_m f_k - b_m e_k \\ N_{km} &= -J_{km} = a_m e_k + b_m f_k \end{aligned} \quad (6a)$$

Terms of (4) where m is equal to k are

$$\begin{aligned} H_{kk} &= -Q_k - B_{kk} E_k^2 \\ L_{kk} &= Q_k - B_{kk} E_k^2 \\ N_{kk} &= P_k + G_{kk} E_k^2 \\ J_{kk} &= P_k - G_{kk} E_k^2 \end{aligned} \quad (6b)$$

Derivations of (6a) and (6b) are given by Nan Vess.⁽⁴⁾ The terms ΔP_k and ΔQ_k are residuals of (2) as defined in (7).

$$\begin{aligned} \Delta P_k &= P_k (\text{scheduled}) - P_k (\text{actual}) \\ \Delta Q_k &= Q_k (\text{scheduled}) - Q_k (\text{actual}) \end{aligned} \quad (7)$$

For a system of N nodes including the slack node and having S nodes with fixed voltage magnitudes, there are $(2N - S - 2)$

linear equations similar to (3). There are no equations for the slack node, but its effect enters the system through the terms H_{kk} , J_{kk} , N_{kk} , and L_{kk} of equations for nodes that are connected to it. The pattern of nonzero elements of the Jacobian matrix is the same as that of the system admittance matrix if the elements of the Jacobian are taken to be its 2-by-2, 1-by-2, 2-by-1, or 1-by-1 submatrices. It is symmetric in pattern but unsymmetric in value.

Iterative Algorithm

With programming details omitted, the basic iterative algorithm for solution of an unadjusted power flow problem by Newton's method is as follows:

1) An initial approximation to the voltage solution of (2) is assigned. One way is to set the voltage magnitudes, where given, to their given values and to set the other voltage magnitudes equal to that of the slack node. The use of a per unit system is assumed. All angles are set equal to the slack node angle. This will be referred to as a flat voltage start.

2) One cycle of the method of successive displacements is performed without over-correction to assure a favorable start.

3) The Jacobian matrix (3) is formed and augmented by the column of residuals.

4) The voltage corrections are solved by Gaussian elimination and back-substitution. This operation transforms the Jacobian into an upper triangular matrix and the augmented column of residuals becomes a column of voltage angle and magnitude corrections in polar form. The corrections are applied to the node voltages.

5) The residuals ΔP_k and ΔQ_k are checked. If they are sufficiently small, the problem is solved. If not, the procedure is repeated, starting with step 3.

The dimensions of large power system problems are such that a straight-forward implementation of this algorithm would be grossly inefficient, if not impossible. However, because of the extreme sparsity of equations based on nodal formulation, direct solutions can be obtained efficiently by optimally ordered elimination and special programming techniques. These two topics are considered next.

PROGRAMMING

Programming techniques are usually omitted in presentations of solution methods. In this case, however, where programming can make the difference between the method's success or failure, it is necessary to give it some attention. The techniques to be outlined are not necessarily the best, but they have proved reasonably effective and could be considered the point of departure for the development of better ones. Further improvements are certainly possible. A simplified flow chart of the iterative algorithm is shown in Fig. 2.

Triangularization of Jacobian Matrix

A matrix is usually triangularized by eliminating elements of successive columns below the diagonal. From the standpoint of both speed and storage it is more efficient to eliminate elements of each row up to the diagonal before proceeding to the next row. This is shown schematically in matrix (8).

$$\begin{bmatrix} 1 & X & X & X & X & X \\ & 1 & X & X & X & X \\ & & 1 & X & X & X \\ X & X & X & X & X & X \\ X & X & X & X & X & X \\ X & X & X & X & X & X \end{bmatrix} \quad (8)$$

The situation displayed is after processing of the third row. Rows below the third have not yet been operated upon. Using this scheme, rows can be formed and operated upon one at a time in such a way that the maximum storage requirement will not exceed that for the resultant upper triangle.

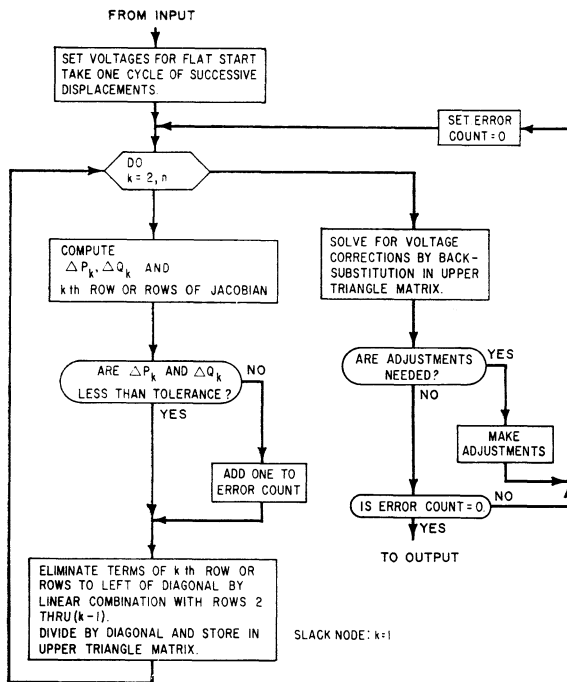


Fig. 2. Simplified flow chart.

Storage of Upper Triangle

Equation (3) is formed and solved by Gaussian elimination and back-substitution. The program operates upon and stores only the nonzero elements. At the start of each iteration the essential data are the initial estimate or latest iterate of node voltages, the nodal admittance matrix, and the scheduled node quantities P_k and Q_k . From these the augmented Jacobian matrix is formed and triangularized in optimum order either one or two rows at a time, depending on whether the corresponding node is of type B or type A , respectively. All elements to the left of the diagonal of a single or double row are eliminated by appropriate linear combinations with previously processed rows, and the elements of the new single or double row are stored in the upper triangle. Matrix (9) shows the form of the upper triangle resulting from Gaussian elimination of the Jacobian matrix (3) augmented on the right by the modified column of residuals.

The primes indicate elements that have been altered from their original values in the elimination process; the unprimed elements are new terms introduced by the elimination process. The situation displayed in (9) is at the end of the downward operations. At this point $\Delta Q_7' = \Delta E_7/E_7$. The remaining voltage and angle corrections can be computed by back-substitution.

1	N_{22}'	H_{23}'		$H_{25}'N_{25}'$			$\Delta P_2'$
1	J_{23}'			$J_{25}'L_{25}'$			$\Delta Q_2'$
	1	$H_{34}'N_{34}'$	$H_{35}'N_{35}$	H_{36}'	$H_{37}'N_{37}'$		$\Delta P_3'$
		1	$H_{45}'N_{45}'$	H_{46}	$H_{47}'N_{47}'$		$\Delta P_4'$
		1	$J_{45}'L_{45}'$	J_{46}	$J_{47}'L_{47}'$		$\Delta Q_4'$
			1	N_{55}	H_{56}'	$H_{57}'N_{57}'$	$\Delta P_5'$
			1	J_{56}'	$J_{57}'L_{57}'$		$\Delta Q_5'$
				1	$H_{67}'N_{67}'$		$\Delta P_6'$
					1	N_{77}'	$\Delta P_7'$
					1		$\Delta Q_7'$

(9)

The scheme for storing this matrix in the computer memory is illustrated in Tables II and III. The primes are omitted in this display. The integers in the upper triangle table represent node numbers corresponding to columns of the triangle and the alpha symbols represent elements of the triangularized Jacobian matrix. The negative sign on node numbers identifies type B nodes. The logic of the storage and indexing scheme will become apparent from a careful comparison of (9) with Tables II and III.

Table III is an index by node number of the starting locations in Table II of entries for each single or double row of (9). If the sign of the entry in Table III for node k is positive, the first three entries starting in the indicated location in Table II are ΔP_k , ΔQ_k , and N_{kk} . These are followed by a signed column indicator m ; if m is positive, it is followed by H_{km} , J_{km} , N_{km} , and L_{km} ; if m is negative, it is followed by H_{km} and J_{km} . These terms are followed by another column entry and the scheme repeats to the end of the row. Similarly, if the sign for the location of node k is negative, the first entry in Table II is ΔP_k followed by a column indicator m ; if m is positive, it is followed by H_{km} and N_{km} ; if negative, by H_{km} only. The end of row k is implied by the starting location of node $(k+1)$ in Table III. There is no entry for $k=1$, the slack node.

Tables II and III are rebuilt every iteration. It might be thought that at least Table III and the integer entries of Table II would remain fixed, but this is not true when node types change as they do in adjusted solutions. Diversity is taken advantage of by using only one table (Table II) for several kinds of entries rather than reserving separate tables for the maximum possible number of entries of each kind.

This scheme is a compromise between conflicting speed and storage requirements. It could be made more compact by storing several integers in one word, or it could be made faster by using separate tables for the residuals, Jacobian elements, and column tags. Obviously, the choice of scheme depends on the relationship between problem size and computer memory.

Working Row Scheme

In addition to a compact storage scheme for the upper triangle, it is advantageous to use a compact working row in which to perform the elimination operations. Because of optimal ordering, the number of nonzero columns remains very small until quite near the end of the elimination process, and even then it is less than ten percent of the total number of nodes. Time and storage can be saved by taking advantage of this fact with a scheme such as that given in Table IV. This table shows the initial and final conditions of the working row for processing nodes 3 and 4 of the example problem. A single or double row of the Jacobian matrix is formed and stored with column designations and sequence indicators as shown. In the elimination process the original column entries become modified and new column entries are added. Most of the added columns would normally occur in mid-sequence of the packed row; in this scheme, however, they are added at the end of the row and the sequence indicators (next LOC in Table IV) are changed accordingly. When all columns to the left of the diagonal have been eliminated, and the diagonal, in effect, has been transformed to unity, the row is transferred to Table II in proper column sequence along with column indicators and the modified residuals. Then the appropriate count of terms is added to the index in Table III to indicate the starting location for the next row.

This scheme saves both time and storage. Since the initial entry is always a store operation, zeroing is unnecessary. All operations on a column are completed before going to the next column, thereby saving indexing operations.

OPTIMALLY ORDERED ELIMINATION

It is necessary to arrange the equations of the Jacobian in an order that will tend to minimize the accumulation of nonzero terms in the upper triangular matrix during Gaussian elimination.

TABLE II
STORAGE SCHEME FOR UPPER TRIANGLE TABLE

LOC	Item	LOC	Item	LOC	Item	LOC	Item	LOC	Item	LOC	Item
1	ΔP_2	11	L_{25}	21	7	31	L_{45}	41	ΔQ_5	51	ΔP_6
2	ΔQ_2	12	ΔP_3	22	H_{37}	32	-6	42	N_{55}	52	7
3	N_{22}	13	4	23	N_{37}	33	H_{46}	43	-6	53	H_{67}
4	-3	14	H_{34}	24	ΔP_4	34	J_{46}	44	H_{56}	54	N_{67}
5	H_{23}	15	N_{34}	25	ΔQ_4	35	7	45	J_{56}	55	ΔP_7
6	J_{23}	16	5	26	N_{44}	36	H_{47}	46	7	56	ΔQ_7
7	5	17	H_{35}	27	5	37	J_{47}	47	H_{57}	57	N_{77}
8	H_{25}	18	N_{35}	28	H_{45}	38	N_{47}	48	J_{57}	58	
9	J_{25}	19	-6	29	J_{45}	39	L_{47}	49	N_{57}	59	
10	N_{25}	20	H_{36}	30	N_{45}	40	ΔP_5	50	L_{57}	60	

TABLE III
INDEX TO UPPER TRIANGLE TABLE

Node	Location of Row In Upper Triangle Table
1	1
2	-12
3	24
4	40
5	-51
6	55
7	58
8	

TABLE IV
CONCEPT OF PACKED WORKING ROW

(A)	LOC	1	2	3	4	5	6
Row 3 before	Column	2	-3	4	-6	7	
elimination of	H	H_{32}	H_{33}	H_{34}	H_{36}	H_{37}	
column 2	J						
	N	N_{32}		N_{34}		N_{37}	
	L						
	Next LOC	2	3	4	5	E	
(B)	LOC	1	2	3	4	5	6
Row 3 after	Column	2	-3	4	-6	7	5
elimination of	H	X	X	H_{34}'	H_{36}'	H_{37}'	H_{35}'
column 2 and	J						
division by H_{33}'	N	X		N_{34}'		N_{37}'	N_{35}'
	L						
	Next LOC	2	3	6	5	E	4
(C)	LOC	1	2	3	4	5	
Double row 4	Column	-3	4	5	7		
before elimina-	H	H_{43}	H_{44}	H_{45}	H_{47}		
tion of column	J	J_{43}	J_{44}	J_{45}	J_{47}		
3	N		N_{44}	N_{45}	N_{47}		
	L		L_{44}	L_{45}	L_{47}		
	Next LOC	2	3	4	E		
(D)	LOC	1	2	3	4	5	
Double row 4	Column	-3	4	5	7	-6	
after elimina-	H	X	X	H_{45}'	H_{47}'	H_{46}'	
tion of column	J	X	X	J_{45}'	J_{47}'	J_{46}'	
3 and J_{44} , divi-	N		N_{44}'	N_{45}'	N_{47}'		
sion of H and	L		X	L_{45}'	L_{47}'		
N terms by H_{44} ,	Next LOC	2	3	5	E	4	
and division of							
J and L terms							
by L_{44}' .							

Note: Display based on sample problem of Fig. 1 and matrices (3) and (9). X means element effectively eliminated or made equal to unity. E designates highest column of the row. Blank locations may contain garbage.

(The example problem is not optimally ordered.) The order needs to be determined only once for each network configuration no matter how many times the solution is iterated.

Methods of Optimal Ordering

The order must be determined prior to elimination. The only information needed by the subroutine for doing this is a table describing the node-branch connection pattern of the network. An order that would be optimal for the reduction of the admittance matrix of the network is also optimal for the triangularization of the related Jacobian matrix.

No method other than complete exhaustion of all possibilities has been found for determining the absolute optimum order. Three schemes for determining approximate optimums have been used. They are listed in increasing order of efficacy and execution time.

1) The nodes are numbered, starting with that having the fewest connected branches and ending with that having the most connected branches. This method does not take into account anything that happens during the elimination steps, but it is simple to program and fast to execute.

2) The nodes are numbered so that at each step of the elimination the next node to be eliminated is the one having the fewest connected branches. This method requires simulation of the elimination process to take into account the changes in the node-branch connections effected at each step.

3) The nodes are numbered such that at each step of the elimination the next node to be eliminated is the one that will introduce the fewest new equivalent branches. This requires simulation of every feasible alternative at each step.

The choice of scheme is a trade-off between speed of execution and number of times the result is to be used. For Newton's method of power flow solution, scheme 2 seems best. Scheme 3 has not proved sufficiently better to offset the increased time required for its execution. Scheme 1 is good for problems requiring only a single solution with no iterations. If the number of iterations for solution of the power flow problem were greater, scheme 3 would probably be preferred.

Examples of Optimal Ordering

Some effects of ordering by scheme 2 are shown in Table V. The indicated number of equivalent branches in the reduced network is the same as the number of nonzero terms that would be developed in the triangularized admittance matrix and is a direct measure of the effectiveness of renumbering. From the standpoint of the power flow solution the figure of importance is the approximate number of terms in Table II. Since this number is a function not only of the number of equivalent branches in the reduced network but also of the number and connection pattern of the type A and B nodes, the figures shown are representative, but not indicative, of an exact relationship. It should be noted that the 949-node problem requires less than twice the

TABLE V
EFFECTS OF OPTIMALLY ORDERED ELIMINATION ON
TYPICAL NETWORKS

Number of Nodes in Original Network	Number of Branches in Original Network	Number of Equivalent Branches of Reduced Network	Approximate Number of Entries in Table II
26	30	45	225
277	433	1097	5248
326	509	1102	5828
350	553	1282	6725
487	711	1274	6646
949	1361	2453	12 920

Note: Execution time for determining order approximately proportional to number of nodes. Approximately ten seconds required for 949-node example on IBM 7040.

memory of the 487-node problem, the 277-node problem requires almost as much memory as the 487-node problem, and the 350-node problem requires more memory than the 487-node problem.

These results may appear to contradict the statement that memory and time requirements increase approximately in direct proportion to problem size, since this relationship holds only for the 487- and 949-node examples. The main reason the number of terms in the triangularized matrix tends to become proportional to problem size as problems become larger is that large networks are composed of subnetworks with relatively few interconnections and the optimal ordering scheme takes advantage of this property. For example, the number of terms in the triangularized matrix of a network composed of two subnetworks with one interconnection would be equal to the sum of the terms in the triangularized matrices of the subnetworks plus one. A detailed analysis of the effects of optimal ordering will be given in another paper.

ADJUSTABLE FEASIBLE SOLUTIONS

To eliminate cut and try procedures, the power flow program must be able to monitor tentative solutions to assure that they fulfill certain given conditions and do not violate constraints. If the tentative solutions are not feasible from the standpoint of these criteria, the program must automatically change the conditions of the problem and repeat the solution until an acceptable one is obtained or found to be impossible. This is quite easy to do with Newton's method.

Adjustment Features of BPA Program

The following automatic adjustments for feasible solutions have been included in the BPA program:

- 1) The taps of any transformer may be adjusted within limits to hold the voltage at any specified node to a given value or the reactive flow through the transformer to any given value.
- 2) The angle of any phase shifting transformer may be adjusted within its limits to regulate the flow of real power through it.
- 3) A type A node may be checked for voltage constraints. If the given Q_k would result in violation of a voltage constraint, the node type is changed from A to B to hold the relevant constraint.
- 4) A type B node may be checked for reactive constraints. If the given E_k would result in violation of a reactive constraint, the bus type is changed from B to A to hold the relevant constraint.
- 5) The scheduled interchange of real power between designated areas of the network may be obtained by adjusting the power output of designated generators in each area.
- 6) The voltage at a specified type A node may be held by automatic adjustment of the voltage at a remote node. Reactive limits may be imposed at the remote node.

The equations employed in implementing these adjustments

TABLE VI
EFFECT OF ADJUSTMENTS ON SOLUTION OF 487-NODE PROBLEM*

Iteration Number	Sums of Absolute Residuals Before Each Iteration		Nodes with Error 0.001 o/1
	P o/1 Error	Q o/1 Error	
1	375.353	372.218	480
2	63.437	126.187	466
3	10.744	16.482	281
4	0.690	1.279	62
5	0.031	0.193	4
6	0.000	0.001	0

* Same problem as shown in Table I. Adjustments: 30 nodes with voltage or reactive constraints; 2 nodes with remote voltage control.

TABLE VII
EFFECT OF ADJUSTMENTS ON SOLUTION OF 340-NODE PROBLEM*

	Iteration Number	Sums of Absolute Residuals Before Each Iteration		Nodes with Error < 0.001 o/1
		P o/1 Error	Q o/1 Error	
<i>Unadjusted Solution</i>	1	210.543	126.779	337
	2	12.503	24.479	326
	3	0.503	0.501	127
	4	0.001	0.001	0
<i>Adjusted Solution</i>	1	210.543	126.779	337
	2	12.503	24.479	326
	3	1.297	6.222	142
	4	17.752	3.924	31
	5	4.175	4.009	25
	6	0.896	0.721	9
	7	0.663	0.388	4
	8	1.333	0.322	4
	9	0.102	0.017	2
	10	0.000	0.001	0
	11	0.036	0.683	15
	12	0.000	0.001	0

* Problem of 340 nodes, 528 branches. One per unit power equals 100 MVA. Adjustments: 3 area interchanges, 14 tap changing transformers, 1 phase shifting transformer, 48 nodes with reactive constraints, 2 nodes with voltage constraints, 2 nodes with remote reactive control.

are similar to those described for use with iterative methods. Required adjustments are made at the end of the second iteration and at the end of every iteration thereafter until the final adjusted feasible solution is obtained. After a constrained variable has reached a limit it is checked every iteration to see if the limit is still necessary. All adjustments are applied simultaneously and the effect of their interaction is not detected until the end of the next iteration.

Effect of Adjustments on Convergence

The effect of adjustments on the convergence of two representative problems is shown in Tables VI and VII. Table VI shows the 487-node problem of Table I with some constraint adjustments and remote voltage control. Adjustments were made at the end of the second, third, and fourth iterations, after which all constrained variables were within limits and the remotely controlled voltages were within a small tolerance of the desired value. The problem was solved at the end of the fifth iteration but this was not detected until the end of the sixth.

Table VII shows the effect of more extensive adjustments on a problem of 340 nodes and 528 branches. Without adjustments this problem solves in three iterations, but has to run for four. The adjusted solution requires twelve iterations. Tap changing

TABLE VIII
EFFECT OF REUSE OF TRIANGULARIZED JACOBIAN MATRIX ON RATE OF CONVERGENCE*

Iteration	Basic Method		Sums of Absolute Residuals Before Each Iteration			
	<i>P</i> o/1	<i>Q</i> o/1	One Cycle of Reuse		Two Cycles of Reuse	
	<i>P</i> o/1	<i>Q</i> o/1	<i>P</i> o/1	<i>Q</i> o/1	<i>P</i> o/1	<i>Q</i> o/1
1	259.712 46	175.889 75	259.712 46	175.889 75	259.712 46	175.988 75
2	13.641 00	35.202 00	10.607 18	8.933 58	2.653 43	3.664 37
3	1.337 00	1.730 00	0.244 43	0.242 89	0.000 72	0.000 55
4	0.016 00	0.015 00	0.002 73	0.005 48		
5	0.000 24	0.000 19	0.000 03	0.000 05		

* Problem of 286 nodes and 443 branches. One per unit power equals 100 MVA. Reuse cycles occur at end of each basic cycle.

transformers are treated as having continuously adjustable ratios until a tentative solution is reached; then the ratios are changed to the nearest discrete step and no further changes in ratio are made. The problem is then solved again under this condition with other adjustments and constraints still in effect. In the example, a tentative solution was obtained at the end of the ninth iteration, detected at the end of the tenth, after which transformer taps were fixed, and two more iterations were required for a final solution. Contemplated modifications are expected to substantially reduce the number of iterations required for adjusted solutions.

ACCELERATION

Repeated Use of Jacobian Matrix

The standard technique for accelerating Newton's method is to reuse the Jacobian matrix of one iteration for several successive cycles without recomputing it.^[14] Ordinarily this is done by computing the residuals, multiplying by the inverse of the Jacobian to obtain the corrections, applying the corrections, computing new residuals, etc. The process is continued until the problem is solved or the decrease in rate of improvement indicates that the Jacobian should be reevaluated at the new operating point. The method works for unadjusted power flow solutions but, of course, it is necessary to use an alternative to the inverse to repeat the operations of the single solution.

The inverse or nodal solution matrix need never be computed to obtain its effect; there is a much better method for preserving the operations of a single solution for repetition. This method, which lacks a definitive name, is discussed in books on numerical analysis^[15] and has been explained in recent papers^{[10], [11], [13]} pointing out its particular suitability for power system problems. Application of this method requires about 40 percent more problem dependent storage, since, in addition to the upper triangle for the back solution, it is also necessary to save the lower triangle for repetition of the forward solution. The actual computer storage scheme, omitted here, is somewhat more complicated than for the upper triangle alone, but equally efficient in execution.

The effectiveness of this method in solving a 286-node, 443-branch problem is shown in Table VIII. With the basic method this problem was solved in four iterations but had to run the usual five until this condition was detected. With one subcycle of reuse after the first, second, third, and fourth iterations, it again took until the end of the fifth iteration to detect that the problem was solved at the end of the fourth. With two subcycles of reuse after the first, second, and third iterations it was found that the problem was solved (sums of absolute residuals less than 0.001 per unit MVA) at the end of the third iteration subcycles. Each subcycle takes approximately 25 percent as long as one basic cycle; therefore, since six subcycles were needed to save two basic cycles, the net saving in time was only about one-half a basic cycle.

Although the Jacobian does not have to be triangularized

again, a subcycle takes a relatively long time to execute because the residuals have to be computed and the square root, sine, and cosine routines have to be used to convert polar to rectangular corrections. It is evident that this method of acceleration would be more effective in the rectangular formulation where the ratio of subcycle to main cycle time would be smaller. In the polar formulation it does not appear to offer enough advantage to be considered when memory limitations are critical. It is expected to be useful in extensions of the method to related problems and it suggests that the rectangular version of the method should be considered when larger computer memories become available.

Intermixed Successive Displacements

It has also been found that subcycles of successive displacements between the basic cycles of Newton's method reduce the total number of basic cycles, but again the total time remains about the same. Other possibilities for acceleration remain to be explored, but, fortunately, the method is already quite fast.

INITIAL APPROXIMATION

A favorable starting approximation is necessary for a successful solution. With an unfavorable start the method will diverge or converge to an unfeasible solution. Ordinarily, the flat voltage start is sufficient. Voltage magnitudes are set equal to their scheduled values or, if not scheduled, equal to the magnitude of the slack node voltage, and all angles are set equal to the slack node reference angle. There are some situations, however, for which this is not a sufficiently good start. It has been found that the flat start followed by one cycle of successive displacements without overcorrection is consistently favorable.

As would be expected, a more accurate initial approximation reduces the number of iterations required for solution. In a series of related problems the previous solution is usually a good start for the next problem. However, since the method converges rapidly from a flat start and checking for adjustments can take place after only two iterations, it is hardly worthwhile using any other starting method. Using the voltages of a related case would normally save only one iteration. In the BPA program every solution uses the flat start followed by one cycle of successive displacements.

RELATED DEVELOPMENTS

The main intent of this paper has been to present convincing evidence of the computational feasibility and advantages of Newton's method for the power flow problem. However, because this method is based on the Jacobian matrix of the nodal power equations, it leads quite naturally to other applications. The Jacobian matrix provides a linearized mathematical model of the power system at any given operating point. If this matrix is triangularized by Gaussian elimination, and the forward as well as the usual back solution is retained, the pair of triangular matrices can be used to obtain the effect of the Jacobian, its inverse, the transpose Jacobian, and its inverse. All these ma-

trices are useful in other applications. This will be explained in detail in another paper, but it follows directly from articles already published.^{[10], [13], [15]} It is pointed out here to indicate the anticipated trend of further developments and as an additional inducement for using the method.

Preliminary tests indicate that Newton's method should be very effective in the digital transient stability program as a means of solving for voltages at nodes with constant P and Q . In this application it would be advantageous to preserve the forward and back solutions of the Jacobian for reuse for several time steps of the transient. Since all the nodes in this case would be type A , there would appear to be a definite advantage in using the rectangular form of the method for this particular application. This form requires no more memory than the polar form for type A nodes and it avoids use of the square root and trigonometric routines which account for much of the time burden of the polar form. Unfortunately, because of memory limitations Newton's method has not been used in any form in the present BPA transient stability program which must allow for 500 nodes with possible constant P and Q . Optimally ordered triangular factorization is used in conjunction with another less efficient direct method of power flow solution.

Recently it has been learned that Electricité de France is following a somewhat different approach to the use of Newton's method in the power flow and related optimization applications.^{[18], [19]} Insufficient information is available for a meaningful comparison of approaches at this time.

CONCLUSION

By the use of optimally ordered Gaussian elimination and special programming techniques, the ac power flow problem can be solved efficiently by Newton's method, a standard technique for nonlinear equations. This method appears to be faster, more accurate, and more reliable than any other known method for any size or kind of problem. Because of computer memory restrictions, present experience has been limited to the polar formulation of this method; an alternative rectangular formulation, having identical convergence properties, might prove more advantageous when computer memory is not critical. Although the method is already quite good, it is evident that further improvements are possible. An additional inducement for use of the method is its adaptability to other applications. The important contribution of this paper is the evidence that the method not only works, but that it is efficient for large problems.

REFERENCES

- [1] J. B. Ward and H. W. Hale, "Digital computer solution of power flow problems," *Trans. AIEE (Power Apparatus and Systems)*, vol. 75, p. 398, June 1956.
- [2] R. J. Brown and W. F. Tinney, "Digital solutions for large power networks," *Trans. AIEE (Power Apparatus and Systems)*, vol. 76, p. 347, June 1957.
- [3] A. F. Glimm and G. W. Stagg, "Automatic calculation of load flows," *Trans. AIEE (Power Apparatus and Systems)*, vol. 76, p. 817, October 1957.
- [4] J. E. Van Ness, "Iteration methods for digital load flow studies," *Trans. AIEE (Power Apparatus and Systems)*, vol. 78A, p. 583, August 1959.
- [5] —, "Convergence of iterative load flow studies," *Trans. AIEE (Power Apparatus and Systems)*, vol. 78B, p. 1590, February 1959.
- [6] H. W. Hale and R. W. Goodrich, "Digital computation of power flow—some new aspects," *Trans. AIEE (Power Apparatus and Systems)*, vol. 78A, p. 919, October 1959.
- [7] J. E. Van Ness and J. H. Griffin, "Elimination methods for load flow studies," *Trans. AIEE (Power Apparatus and Systems)*, vol. 80, p. 299, June 1961.
- [8] H. E. Brown, G. K. Carter, H. H. Happ, and C. E. Person, "Power flow solution by matrix iterative method," *Trans. AIEE (Power Apparatus and Systems)*, vol. 82, p. 1, April 1963.
- [9] M. A. Laughton and M. W. Humphrey Davies, "Numerical techniques in solution of power-system load-flow problems," *Proc. IEE (London)*, vol. 111, p. 1573, September 1964.
- [10] N. Sato and W. F. Tinney, "Techniques for exploiting the sparsity of the network admittance matrix," *IEEE Trans. Power*

Apparatus and Systems, vol. 82, p. 944, December 1963.

[11] L. Carpentier, "Ordered eliminations," *Proc. 1963 Power System Computation Conf.*

[12] R. Baumann, "Some new aspects of load flow calculation," *Proc. 1965 Power Industry Computer Applications Conf.*, p. 91.

[13] H. Edelman, "Ordered triangular factorization of matrices," *Proc. 1966 Power System Computation Conf.*, sec. 4.12.

[14] C. G. Broyden, "A class of methods for solving non-linear simultaneous equations," *Mathematics of Computation*, vol. 19, p. 577, 1965.

[15] A. Ralston, *A First Course in Numerical Analysis*. New York: McGraw-Hill, 1965, p. 408.

[16] G. Hadley, *Nonlinear and Dynamic Programming*. Reading, Mass.: Addison-Wesley, 1964, p. 48.

[17] A. S. Householder, *Principles of Numerical Analysis*. New York: McGraw-Hill, 1953, p. 135.

[18] D. Feingold and D. Spohn, "Bounded variables load flow problems," *Proc. 1966 Power System Computation Conf.*, sec. 4.3.

[19] —, "Direct methods for linear networks of load flow computation," *Proc. 1966 Power System Computation Conf.*, sec. 4.4.

[20] J. Peschon, W. F. Tinney, D. S. Piercy, and O. J. Tveit, "Optimum control of reactive power flow," presented at 1967 IEEE Winter Power Meeting, New York, N. Y., January 29–February 3.

Discussion

Hermann W. Dommel (Bonneville Power Administration, Portland, Ore.): The authors have presented a method which improves the solution of power flow problems considerably in speed and in accuracy. They have applied two fundamental ideas: the use of Newton's method and the solution of the resulting set of linear equations by means of optimally ordered Gaussian elimination with a compressed storage scheme. Both are a great step forward in the use of digital programs for power flow solutions. The data presented in the paper clearly demonstrate the impressive improvements.

Basically, there are three nodal methods used for the solution of power flow problems, all iterative:

1) The nodal iterative method, using the admittance matrix and an iterative process similar to the Gauss-Seidel iterative solution of linear equations.

2) The impedance matrix iterative method, using the impedance matrix (or a hybrid matrix or a triangularized admittance matrix). It is equivalent to the direct solution of the linearized system with given node currents, whose values are changed from step to step to get the scheduled power values. The matrix is complex and usually symmetric; it remains unchanged for all steps.

3) Newton's method as described in this paper, using a Jacobian matrix (in Europe also called functional matrix), where in each step a system of linear equations has to be solved and the matrix (real, nonsymmetric) changes after each step.

It is very difficult to make a theoretical comparison of these different methods; therefore, one has to rely more or less on experience gained from solving actual problems. Some of the typical features of the three methods may be seen in Figs. 3 and 4, which were obtained from the solution of a 17-node problem.^[1] Figure 3 shows a plot of the real and imaginary part of the voltage of some node as a function of the iteration step. Newton's method leads to the final solution with the least number of iteration steps and with a very steep descent. A steep descent into the final solution is essential in iteration processes for achieving small truncation errors. A helpful estimate for the truncation error $|x^{(h)} - x|$, ($x^{(h)}$ = value at step h , x = exact but unknown value), is given by Baumann^[3]:

$$|x^{(h)} - x| \leq \frac{k}{1 - k} |x^{(h)} - x^{(h-1)}|$$

Manuscript received February 20, 1967.

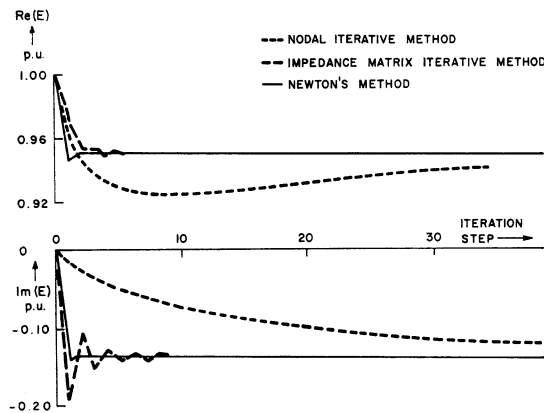


Fig. 3. Voltage as function of iteration steps.

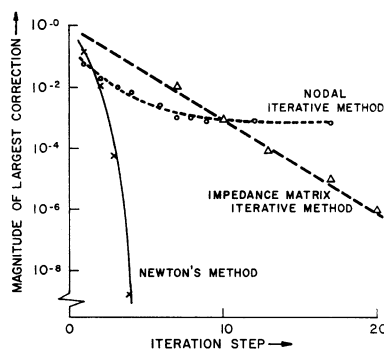


Fig. 4. Voltage correction as function of iteration steps.

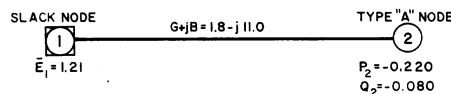


Fig. 5. Simple network.

with the convergence factor k being the smallest upper bound for

$$k_i = \frac{x^{(i+1)} - x^{(i)}}{x^{(i)} - x^{(i-1)}} \text{ with } i > h.$$

For functions of more than one variable, k would be the upper bound for the highest magnitude of the eigenvalues of the Jacobian matrix. Only when $k < 0.5$ can one expect the truncation error to be less than the magnitude of the latest correction. This is usually true for the impedance matrix iterative method and even more so for Newton's method. In the nodal iterative method, however, k can approach one (with or without the use of relaxation factors [3]), which means that the truncation error may be large even when the latest correction is very small.

For the 17-node problem of Figs. 3 and 4, a termination with the latest corrections $\leq 10^{-4}$ gave an approximate solution which, in the nodal iterative method, was still off by 15 percent for the total losses, while Newton's method and also the impedance matrix iterative method yielded results which were exact for any practical purpose. The convergence properties of the three methods can be seen by plotting the magnitude of the largest correction as a function of the iteration step (Fig. 4); Newton's method has quadratic convergence, the impedance matrix iterative method has linear convergence.

From the standpoint of convergence, Newton's method is clearly the best. Unfortunately, one has to solve a large system of linear equations in each step with the matrix changing from step to step. For this reason the discussor discarded it in 1962 as a practical method and has used it only when solving optimization problems, where the transpose of the Jacobian matrix must be used anyway, to find Lagrange multipliers.^[2] The authors must be commended for overcoming this computational difficulty by using optimally ordered elimination and thus developing Newton's method into a very powerful and practical technique.

The authors mention the possibility of using rectangular coordinates as variables instead of polar coordinates. Van Ness showed that the convergence behavior will basically be the same with rectangular coordinates, provided one is sufficiently close to the solution.^[4] Some experiments were carried out with a 26-node and a 279-node problem, using rectangular coordinates for nodes with P, Q specified and polar coordinates for nodes with P, E specified (it seems natural to retain the polar coordinate δ as the variable for the latter; otherwise, the number of equations would increase); this version needed two to three additional steps to reach the same accuracy. Experience seems to indicate that the version with rectangular coordinates might converge slightly better for problems where P, Q are specified at all or almost all nodes.

Interestingly enough, there is still another version possible, which the discussor has used.^{[1],[2]} Equation (2) of the paper could also be written

$$\frac{P_k + jQ_k}{\bar{E}_k} - \sum_{m=1}^N \bar{Y}_{km} \bar{E}_m = 0 \quad (10)$$

which might be called the node-current form. Now, breaking (10) down into two real equations and applying Newton's method is slightly different from applying Newton's method to the two real equations corresponding to (2) of the paper. In both cases P_k and Q_k are regarded as specified. In this node-current version the off-diagonal terms for nodes with P_k, Q_k specified are simply terms from the admittance matrix,

$$H_{km} = -L_{km} = B_{km}$$

$$N_{km} = J_{km} = G_{km}.$$

This version seemed better for problems with no loads or very light loads; it can be shown that if all P_k, Q_k approach zero then the Jacobian matrix becomes the admittance matrix (in the real form) and the first step then yields the exact solution of the linear problem.

Since it might seem strange that the choice between polar and rectangular coordinates as well as between (2) and (10) influences the convergence, (the eigenvalues of the Jacobian matrix would show it, of course), numerical values for the simplest problem of Fig. 5 are given in Table IX. For the general case it is hard to predict which version is best; experiments showed however, that the method described in the paper seems to be best for the normal case with P, Q

TABLE IX
VERSIONS OF NEWTON'S METHOD FOR NETWORK IN FIGURE 5

Newton's Method Applied to	\bar{E}_2 Treated as Two Real Variables with	First Step	Second Step
Equation (2)	polar coordinates	1.252 68 - j0.018 97	1.203 31 - j0.015 26
Equation (2)	rectangular coordinates	1.252 82 - j0.015 14	1.203 31 - j0.015 14
Equation (10)	polar coordinates	1.201 41 - j0.017 40	1.201 26 - j0.015 14
Equation (10)	rectangular coordinates	1.201 53 - j0.014 48	1.201 26 - j0.015 14

Note: \bar{E}_2 —initial guess = 1.0; final value = 1.201 259 - j0.015 140.

specified on most nodes and P , E specified on few nodes. For certain extraordinary cases, e.g., no load, another version might be better and yield the desired feasible solution, where otherwise the program might not converge at all or arrive at a physically unfeasible, though mathematically correct, solution.

The differences in convergence between using polar coordinates (differentiation in the real domain with respect to magnitude and angle of voltages) and rectangular coordinates (differentiation with respect to real and imaginary part of voltages) is due to the fact that the complex power equation (2) is not an analytic function of the complex voltage because of conjugate terms. Then the Cauchy-Riemann differential equations are not fulfilled and no first complex derivative exists; the rate of change of the function depends on the path.

REFERENCES

- [1] H. Dommel, "Digitale Rechenverfahren fuer elektrische Netze," dissertation, Munich Institute of Technology, Germany, 1962.
- [2] —, "Digitale Rechenverfahren fuer elektrische Netze," *Arch. fuer Elektrotech.*, vol. 48, pp. 41–68, February 1963; pp. 118–132, April 1963.
- [3] R. Baumann, "Mathematische Behandlung von Aufgaben der Netzplanung und des netzbetriebs," *Elektrotech. Z. A*, vol. 87, pp. 351–357, May 1966.
- [4] J. E. Van Ness and J. H. Griffin, "Elimination methods for load-flow studies," *Trans. AIEE (Power Apparatus and Systems)*, vol. 80, pp. 299–304, June 1961.

Homer E. Brown (Commonwealth Edison Company, Chicago, Ill.): The authors are to be commended for their persistence in seeking a method of solving power flow problems in large interconnected systems. It is very encouraging to those in the field of network analysis that, in spite of the fact that the nodal iterative method almost completely dominates the field of power flow solutions, research teams are actively seeking methods that may one day replace the nodal iterative method.

The method reported in this paper shares the advantages of the Z -matrix method,¹ i.e., the ability to treat negative impedance values and large impedance mismatch of lines connected to any node, and requires practically the same number of iterations for any system; and, at the same time, overcomes the serious size limitation of that method. Newton's method has the ability to handle systems as large as the nodal iterative method, but does not have the convergence difficulties of the nodal method. Further research and development of the method is therefore highly recommended since it possesses the advantages of both the Y -matrix and Z -matrix methods without their inherent weaknesses.

At the completion of a power flow study, frequently it is desirable to continue with a transient stability study of the system. The nodal iterative method can continue from its converged values and proceed with the swing curve calculation. The nodal iterative method now appears in its most unfavorable light, since the iteration procedure in the step-by-step calculation involves every node of the system. In the Z -matrix method only the generator axes are involved in a direct calculation.² The step-by-step calculation is therefore much faster. Would the authors comment on the transient stability methods available to the Newton method on a *load-and-go* basis? Would the calculation continue to be a voltage solution of a large system or could a direct calculation be made using only the generator buses?

In power flow work it is often required to obtain a flow in which only the generation schedule is changed. Would the authors indicate the amount of data required for a change case of this type? Can base cases be saved on tape for restart or does each new case start from raw data?

Manuscript received February 13, 1967.

¹ H. E. Brown, G. K. Carter, H. H. Happ, and C. E. Person, "Power flow solution by matrix iterative method," *Trans. AIEE (Power Apparatus and Systems)*, vol. 82, p. 1, 1962.

² H. E. Brown, H. H. Happ, C. E. Person, and C. C. Young, "Transient stability by an impedance matrix method," *IEEE Trans. Power Apparatus and Systems*, vol. PAS-84, p. 1204, December 1965.

R. J. Brown (Department of the Interior, Washington, D. C.): Mr. Tinney and his associates are to be congratulated on their unrelenting 10-year assault upon the general power flow solution in all its ramifications. It has been my privilege to observe over the years the steady chipping away of the many stumbling blocks encountered in the search for a fast, dependable, general purpose schema capable of performance on modest computing equipment within the reach of the power engineering community. If the research work on the Newton method described had been confined to the basic Newton algorithm no practical betterment over present procedures would have resulted. The practicality of the authors in this presentation as well as in several previous papers is evident in the systematic intermixing of mathematical projection with a number of judiciously arranged and custom-tailored numerical techniques which in aggregate produce the desired result. To illustrate this point, the following numerical techniques are basic to the success of the scheme:

- 1) optimally ordered elimination
- 2) triangulation by row rather than the traditional column algorithm [matrix (8)]
- 3) the packed storage scheme for the upper triangle (Tables II, and III)
- 4) the packed working row (Table IV)
- 5) use of an initial cycle of successive displacement to condition the Newton start.

In brief, the authors' efforts are most significant in bridging the gap between academic efforts and problem solving. They have successfully penetrated the barrier between the serenity of ivory tower learning and the rough-and-tumble world of daily engineering deadlines and unsolved problems. As guidelines are few, I would like to ask for the authors' remarks on any generalized methods they may have developed for approaching major engineering problems in their decade of effort on the general power flow.

Manuscript received February 10, 1967.

Harold Wood (Southern California Edison Company, Los Angeles, Calif.): The advantages of the direct method of solution to the load flow problem have been known since its introduction in 1961. However, the practical use of the method for large systems was limited, due to the excessive memory requirement and slow computational speed.

By exploiting the sparsity of the admittance matrix, and the use of ordered elimination, the 1963 paper by Mr. Sato and Mr. Tinney^[10] represented a major breakthrough in the solution of power flows. The present paper provides additional information on programming and operating what is considered to be the most superior load flow program in existence today.

At present, the Gauss-Seidel nodal iteration (successive displacement) method is in widespread use. It can handle medium-size systems on small-size computers; and with such computers, its use is mandatory. With larger computers (32K word memory), space is available for the additional 8K of memory required for Newton's method, and it is no longer necessary to tolerate the many disadvantages of the nodal iteration method.

Some of these disadvantages are shown in Table X, where the two methods are compared on typical ill-conditioned systems. Any or all these conditions may be encountered in power system operation. Table X illustrates why it is difficult to compare the two methods. There are many practical situations which cannot be solved by Gauss-Seidel, but can be solved by Newton's method. To our knowledge, there is no instance where the converse is true.

We have used Newton's method for six months. With a Control Data 3400 computer, a typical study consisting of 48 buses and 70 lines takes 73 seconds for program and data read-in, solution and print-out, compared to 81 seconds using Gauss-Seidel. As shown previously, the computing time differential can be increased at will by increasing the ill-conditioning.

We concur with the authors' experiences on starting voltages. Newton's method, in a program which removes one line at a time

Manuscript received February 10, 1967.

TABLE X

Type of Problem	Gauss-Seidel Method	Newton's Method
Heavily loaded systems	Usually cannot solve systems with phase shift beyond 70 degrees	Solves systems with phase shifts up to 90 degrees
Systems containing negative reactance, such as 3-winding transformers or series capacitors	Unable to solve	Solves with ease
Systems with slack bus at a desired location	Often requires trial-and-error to find a slack bus location which will yield a solution	More tolerant of slack bus location
Long and short lines terminating on same bus	Usually cannot solve a system with a long-to-short line ratio at any bus beyond 1000 to 1	Can solve a system with a long-to-short line ratio at any bus of 1 000 000 to 1
Long radial type system	Difficulty in solving	Solves a wider range of such problems
Acceleration factors	Number of iterations depends on choice of acceleration factors	None required

automatically, with print-out of overloaded lines only, takes three minutes and six seconds using a flat voltage start for each case, and takes 26 seconds longer using voltages from the previous case. It is believed that the 11 percent savings in computer time is not worth risking a possible convergence to a wrong solution.

The main disadvantages of Newton's method are: 1) the programming logic is considerably more complex, and 2) the memory requirements dictate a computer with at least 32K memory.

The advantages of Newton's method include greater speed and accuracy, and the ability to solve a wider variety of ill-conditioned systems than ever before. The authors are to be commended for their work in presenting this method to the industry.

W. A. Prebble (The Hydro-Electric Commission, Hobart, Tasmania, Australia): The following comments refer to the program, based on the elimination method suggested by Van Ness and Griffin,¹ developed and tested early in 1966 and now used for solving the Commission's load flow problems.

Using rectangular Cartesian coordinates, i.e.,

$$V = e + jf \quad (11)$$

the network voltage correction equations, written in matrix form, are

$$\begin{bmatrix} \Delta Q \\ \Delta P \end{bmatrix} = \begin{bmatrix} U & W \\ S & T \end{bmatrix} \begin{bmatrix} e \\ f \end{bmatrix} \quad (12)$$

in which there are no rows or columns corresponding to the slack busbar. The values used in the U , W , S and T submatrices are given by equations (14) and (15) of Van Ness and Griffin;¹ for generator busbars the equation for voltage magnitude is included in ΔQ , U and W . Equation (12) is solved by column pivoting, with the main diagonal U and T terms as the pivots.

As, for a particular problem, (12) is set up and solved more than once, it is advantageous to predetermine the order of column pivoting which keeps the number of arithmetic operations (also the storage requirements) to a minimum for this problem. The network connections are examined and a list, establishing the order of column pivots, is set up so that at each stage of the calculation, the pivot is the one in the column with the least number of off-diagonal terms; when several columns satisfy this requirement simultaneously, the first of these is used.

Manuscript received February 28, 1966.

¹ J. E. Van Ness and J. H. Griffin, "Elimination methods for load-flow studies," *Trans. AIEE (Power Apparatus and Systems)*, vol. 80, pp. 299-304, June 1961.

From the network, omitting the slack busbar and the connections to it, indexes are set up for locating the off-diagonal terms of the T matrix. The columns with one off-diagonal term (corresponding to busbars at the end of radial lines) are listed first, because when used as pivotal columns no new terms are introduced into the matrix. Next, the first column with two off-diagonal terms is listed and the result of using it as the pivotal column examined; if this introduces new terms in the matrix, then the indexes are modified. The procedure is repeated with the remaining columns having two off-diagonal terms, then those with three, etc., until all the columns are listed.

Equation (12) is then set up and solved by pivoting on the U and T diagonal terms pairs according to the order list, the indexes giving the storage location of the terms used during the calculation.

It is interesting to note that with rectangular voltage coordinates the U , W , S , and T submatrix terms depend directly on the values of e and f chosen. In a high-voltage system, the choice of $1 + j0$ for the initial busbar voltages makes the U and T diagonal terms the ones of greatest magnitude in their respective columns each time (12) is set up (except for generator busbars where the U diagonal term is initially two and all other terms in this U and W row zero). For low-voltage distribution line and cable networks, in which the resistance may be greater than the reactance, a starting voltage such as $0.6 + j0.8$ causes dominance of the major diagonal terms. However, in practice no advantage has been found for starting with a voltage other than $1 + j0$; the solution is obtained just as quickly and accurately even when the main diagonal is not dominant.

From the experience gained with the program, which is written in Algol for the Elliott 503 computer, it is concluded that:

1) Any load flow problem can be solved accurately, i.e., within 0.005 MW, 0.005 Mvar of the specified real, reactive power at load busbars and 0.005 MW, 0.005 kV of the specified real power, voltage magnitude at generator busbars; for practical problems this requires two or three elimination cycles.

2) The choice of slack busbar is not important; radial type networks with slack busbar at the beginning have been solved successfully.

3) It is faster than the nodal iterative method using linear acceleration.

P. L. Dandeno (Hydro-Electric Power Commission, Toronto, Ont., Canada): The authors have presented examples which give convincing evidence of extremely high accuracies for load flow solutions. There may not be the need for such high accuracy for normal load flow cases; however, this discussor believes that the authors' method could be used to great advantage in stability calculations. This was hinted at in their comments about related developments in the paper.

Manuscript received February 24, 1967.

We doubtless are not alone in realizing that the results of transient stability runs depend on the accuracy of the load flow solution at each calculating interval. Using the Gauss-Seidel (nodal iterative) approach in calculating generator power outputs during transient stability runs, we found that a tolerance of 0.001 p.u. in the real and imaginary parts of the voltage correction terms gave a stable answer. When the voltage correction tolerance was decreased or (tightened) to 0.0001, the stability of the system was indicated to be marginal. In other instances the opposite effect was noted. We can only conclude that extremely high accuracy is desirable in certain types of transient stability studies when, for one reason or another, power limits or critical switching times are being closely determined.

As a corollary to the above statements, extremely high accuracy is expensive to obtain in terms of machine time. This is even more evident when some or all the loads in the power system are assumed to be voltage sensitive in varying degrees. In certain cases a satisfactory solution may be obtained if the loads (P and Q) were allowed to vary as the first power of the voltage. The identical system would not run satisfactorily when constant power and constant reactive power were assumed during the transient stability conditions under investigation.

A method such as that proposed by the authors of this paper would appear to avoid this serious problem, and at the same time retain the various benefits of the nodal iterative approach for step-by-step stability studies.

We look forward with interest to any subsequent work by the authors in this important area of power system analysis.

W. F. Tinney and C. E. Hart: The authors are grateful for the excellent discussions that are a valuable supplement to the paper.

Dr. Dommel, who has recently joined the BPA staff, reports on his past and present work using several different forms of Newton's method. He shows that there is a difference in the rate of convergence among the various formulations of the method, and that this is to be expected because the complex nodal power equation on which the Jacobian matrix is based is not an analytic function. Our statement that the convergence properties of the rectangular formulation would be the same is incorrect. There is need for further experimentation to find which form of Newton's method is best for special applications, such as transient stability. Both programming efficiency and rate of convergence should be taken into account. It appears that our choice of the polar form was fortunate because it seems to be best for our problems and, most likely, for the general case.

Mr. H. Brown asks about the possibility of running a transient stability study directly from a power flow solution. This would be feasible, but the method we use transfers the essential power flow data to a tape for subsequent use in an independent transient stability program. When we implement Newton's method in the transient stability program, it will be a specialized subroutine, differing from the one used in the general power flow solution. Mr. Brown also asks about the possibility of using the method with a reduced network in the transient stability program. This could be done, but unless the number of swing buses is very small compared with the number of nonswing buses, it would not help. Paradoxically, with

optimally ordered elimination, *reducing* the network actually increases the data burden and computing time. Mr. Brown has had considerable success with the Z -matrix method of power flow solution. It appears to us that this method might compete with Newton's method in speed and size if some form of ordered triangular factorization were used instead of the explicit Z matrix. It would require more iterations, as Dr. Dommel's Fig. 3 shows, but the iterations should be faster because the system only has to be triangularized once.

Mr. R. Brown asks about our general approach. If it differs from others, it is because we do not separate methods analysis and programming. We disagree with the idea that programming is subordinate to methods analysis and that these activities should be performed by different specialists. The successful application of Newton's method to the power flow problem is an example of what can be accomplished by a synthesis of methods development and programming techniques. In our opinion, the solution of other power system problems could be improved by a similar approach.

Mr. Wood's comments about his experiences with the polar form of Newton's method mostly confirm our experiences. He indicates an influence on convergence due to reference bus location, an effect we did not investigate. He also indicates that he has encountered some ill-conditioned systems that cannot be solved. Our experience indicates that some modification of the starting procedure or a restriction on the magnitude of voltage corrections would probably make these systems solvable. It is also possible that one of the alternative forms of Newton's method described in Dr. Dommel's comments would be better for a radial system. Mr. Wood notes that his program can solve problems with a ratio of line impedances of 10^6 to 1. This is due not only to the direct solution method but also to the large word size of his computer, which minimizes the effects of round-off error.

Mr. Prebble has used the alternative rectangular formulation of Newton's method to which we referred in the paper. His optimal ordering appears to be the same as our scheme 2. As he notes, with a direct as opposed to an iterative method, the magnitude of the diagonal matrix terms does not influence the convergence rate. Nevertheless, all other considerations being equal, a large pivot element is desirable since it minimizes round-off error. Mr. Prebble indicates that only two or three iterations are required, whereas we have indicated that we usually use four or five. This could be due to the difference in methods, rectangular instead of polar, but this is unlikely. In fact, as Dr. Dommel has found, if there is any difference, it is in favor of the polar form. Part of the difference in number of iterations may be due to differences in solution criteria. Another factor affecting the number of iterations is round-off error. The residuals at the end of each iteration are due not only to the linear approximation, but also to the round-off error of the direct solution. When the problem is large and/or the computer word size is small, it may require additional iterations to liquidate the round-off error.

We fully agree with Mr. Dandeno's comments concerning the need for accuracy in the power flow solution for transient stability simulation. Since Newton's method can achieve any specified degree of accuracy, the only questions concerning its application are which version is best and most efficient. For any given accuracy, Newton's method should be much faster than Gauss-Seidel, but it will take more memory. We are experimenting with it now, but cannot give a definitive answer. We find that the assumed power flow conditions during the transient swing, which often turn out to be insoluble by the Gauss-Seidel method, almost always can be solved by Newton's method. Dr. Dommel's comments on truncation error explain why the Gauss-Seidel method cannot produce accurate results and thus should not be relied upon for transient stability purposes.