STATE ESTIMATION OF POWER SYSTEM

by

A. Zarate, G. E. Mauerberger
and O. J. M. Smith

Memorandum No. ERL-M248

4 June 1968

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

Research sponsored by the United States Department of Interior under Contract 14-03-77816.

CONTENTS

APPENDIX

COMPUTER PROGRAMMING

ABSTRACT

A system state vector of node voltages in a power system can be efficiently calculated from measurements of power and var inputs and loads. The preferred sequence is to use Newton's method for voltage angles from power errors, then Newton's method for angles and voltage magnitudes from power and var errors, and lastly the second order move vector with Fletcher-Powell updating of the inverse Hessian matrix. The second order method minimizes an error function whose components are weighted differences between measurements and corresponding calculated values.

The first two or three terms of the Taylor series expansion of a scalar function of a vector, are used to develop iterative methods that provide convergence to the minimum of the function, if this minimum exists. To obtain the operating set of node voltages of an electric power system, an error function is defined, and then, the developed minimization methods are compared with one another, to find the time taken to reach the minimum point using digital computer. The data employed in the power problem is the admittance matrix of nodes of the system, and the set of real and reactive power at each node.

ACKNOWLEDGEMENT

This research was partially supported by Monash University, Melbourne, Australia, the Computer Center, University of California, Berkeley, the Pacific Gas and Electric Company, San Francisco, and Bonneville Power Administration, Portland, Oregon. Thanks are extended to the Organization of American States for a fellowship awarded to Antonio R. Zarate, to Monash University for a Visiting Senior Research Fellowship in Economics and Engineering awarded to Otto J. M. Smith, and to Barry L. Klein, E.E. student at Monash University for his research contributions. Thanks also to Mr. Fernando J. Gonzalez who suggested the use of second order methods.

Otto J. M. Smith, Professor
Electrical Engineering and Computer Sciences,
University of California
Berkeley, California 94720

Gottfried Mauersberger
824 Sycamore Drive
Palo Alto, California 94303

Antonio Zarate
Topo # 308, Col. Chapultepec
Monterey, N. L., Mexico

INTRODUCTION

In order to obtain the operating state of an electric power system, the digital computer has proved, since last decade, to be the most useful tool. However, the iterative and direct methods developed since then, have not improved very significantly in their speed of convergence to the desired solution.

The methods applied here are derived from the Taylor series expansion of an error function of the voltage magnitudes and voltage angles. In chapter 1, the general theory of minimization, using this approach, is briefly explained. Chapter 2 describes the power problem, and also defines the error function to use in the minimization techniques. Chapter 3 sketches the way the digital computer is used and the different programs employed. The results of these programs are shown graphically plotting RMS per unit error, against time taken by the CDC 6400 computer machine. Finally, the conclusions of this research are contained in chapter 4. For those interested in the computer programming itself, the appendix contains the flow diagrams and the program listings in Fortran IV.

CHAPTER 1

MINIMIZATION OF THE VALUE OF A SCALAR FUNCTION USING MULTIDIMENSIONAL TAYLOR SERIES

1.1 The expansion.

Consider a scalar function f whose value depends on the n-vector of adjustables $\underline{x}$. Then for any given n-vector $\underline{x}_o$, the function f may be expanded using Taylor series to obtain a new value, $f(\underline{x}_o + \underline{\Delta x})$, in the neighborhood $\underline{\Delta x}$. Hence, the new value of the function may be expressed as follows:

$$f(\underline{x}_o + \underline{\Delta x}) = f(\underline{x}_o) + \underline{g}'\underline{\Delta x} + \tfrac{1}{2}\underline{\Delta x}'\underline{G\Delta x} + \ldots. \qquad (1\text{-}1)$$

where $\underline{g}'$ is the transpose of an n-vector whose elements are given by $g_i = \partial f/\partial x_i$, and evaluated at $\underline{x}_o$; $\underline{G}$ is an nxn matrix whose elements are $g(i,j) = \partial^2 f/\partial x_i \partial x_j$, and also evaluated at $\underline{x}_o$.

The approximations to be used include only the first three terms, hence, higher order terms are not shown. Note that these terms would include tridimensional matrices and higher.

If it is desired to minimize the value of the function f, then $\underline{\Delta x}$, the vector of corrections, should be chosen in such a manner that the new value, $f(\underline{x}_o + \underline{\Delta x})$, decreases as much as possible. To fulfill the above aim, iterative algorithms may be developed as shown below.

1.2 Steepest descent.

From the first two elements of the expansion, i.e.

$$f(\underline{x}_o + \underline{\Delta x}) = f(\underline{x}_o) + \underline{g}'\underline{\Delta x} \qquad (1\text{-}2)$$

we have,

$$\Delta f = \underline{g}'\underline{\Delta x} \qquad (1\text{-}3)$$

where $\Delta f = f(\underline{x}_o + \underline{\Delta x}) - f(\underline{x}_o)$.

In order to obtain the minimum value of f in the neighborhood $\underline{\Delta x}$, Δf must be as negative as possible. This is accomplished by choosing $\underline{\Delta x}$ in such a way that the inner product, $\underline{g}'\underline{\Delta x}$, is minimized, i.e.

$$\underline{\Delta x} = -\lambda \underline{g} \tag{1-4}$$

The gain λ is necessary because in the general case f is not a linear function of adjustables. In the special case where f is linear, $\lambda = (f(\underline{x}_0) - f_{min})/\underline{g}'\underline{g}$. The optimal value of the gain is found by performing a one-dimensional search using the actual function f.

1.2.1 One-dimensional search.

The search for the best gain λ, is performed in the function f vs. λ that results from the addition of the vector of corrections, $\underline{\Delta x}$, to the vector of adjustables, $\underline{x}$, using differents values for λ in equation (1-4). The values of f are computed at each time.

The question arises as to whether we can locate the minimum value in a more efficient fashion than a direct examination of possibilities. It is necessary to introduce the concept of a unimodal function.

- Definition. A function f(x) is unimodal in an interval (0,b) if there is a number x_0, $0 \leq x_0 \leq b$, such that f(x) is either strictly increasing for $x \leq x_0$, and strictly decreasing for $x > x_0$, or else strictly increasing for $x < x_0$, and strictly decreasing for $x \geq x_0$.

A function with a minimum in an interval (0,b) also satisfies the definition of unimodal function, provided we make a change in sign. It is clear that the unimodal property will never allow an exact evaluation of the minimum value of f(x), but it will allow us to determine very accurately the location of x_0. For this purpose the following theorem is needed:

- Theorem. Let y = f(x) be a unimodal function in an interval

$0 \leq x \leq L_n$. Suppose that L_n is a number with the property that the point at which f(x) achieves its minimum can be located within an interval of unit length by calculating at most n values and making comparisons. Introduce the quantity

$$F_n = \text{Sup } L_n \qquad (1\text{-}5)$$

then,

$$F_n = F_{n-1} + F_{n-2} \qquad n \geq 2 \qquad (1\text{-}6)$$

with $F_o = F_1 = 1$.

The numbers F_n are the Fibonacci numbers which occur in the most unexpected places. In the present discussion this theorem is not proved, however the reader is referred to the bibliography for the proof [3] .

Now, compute $y_1 = f(x_1)$, $y_2 = f(x_2)$, for x_1 and x_2 two values in $(0,L_n)$, to be determined subsequently, with $x_1 < x_2$. If $y_1 > y_2$, the minimum value occurs in (x_1,L_n), while if $y_2 > y_1$, the minimum is in $(0,x_2)$. If $y_1 = y_2$, we choose either of these intervals, even though it is known the minimum occurs in (x_1,x_2).

The problem is then how to determine the subinterval (x_1,x_2) from the initial interval $(0,L_n)$, and all subsequent intervals.

To determine the analytic form of F_n, observe that r^n is a solution of (1-5), ignoring boundary conditions, if

$$r^2 = r + 1 \qquad (1\text{-}7)$$

the two values of r are $(1 \pm \sqrt{5})/2$. Hence, if we set

$$F_n = C_1\left(\frac{\sqrt{5} + 1}{2}\right)^n + C_2\left(\frac{1 - \sqrt{5}}{2}\right)^n \qquad (1\text{-}8)$$

we have the solution of (1-5) provided we fit the values at $n = 0$ and $n = 1$. Thus

$$1 = C_1 + C_2 \tag{1-9}$$

$$1 = C_1\left(\frac{\sqrt{5}+1}{2}\right) + C_2\left(\frac{1-\sqrt{5}}{2}\right) \tag{1-10}$$

the two values for C_1 and C_2 are $C_1 = \frac{1+\sqrt{5}}{\sqrt{5}}$, and $C_2 = \frac{\sqrt{5}-1}{\sqrt{5}}$. Since $(\sqrt{5}+1)/2 > 1$ and $(\sqrt{5}-1)/2 < 1$, it is seen that for large n, F_n is very accurately given by $\left(\frac{1+\sqrt{5}}{\sqrt{5}}\right)\left(\frac{\sqrt{5}+1}{2}\right)^n$. It follows that one can obtain exponential increases in accuracy at a cost of a few additional computations.

The ratio F_n/F_{n-1} approaches $(\sqrt{5}+1)/2$ which is approximately 1.62. Hence, the first two values, x_1 and x_2, should be chosen at a distance 0.62L from either end of the interval (0,L). This uniform testing policy would be an excellent approximate policy for all except the last few stages, at which point it is of little import whether the best policy is used or not. This simple search policy is particularly applicable for use with digital computers.

From above it follows that the problem of locating the minimum reduces to finding the initial interval $(0,L_n)$.

1.3 <u>The second order move vector</u>.

In this method the minimization is accomplished by using the first three terms of the Taylor series expansion shown in (1-1). Hence,

$$\Delta f = \underline{g}'\underline{\Delta x} + \tfrac{1}{2}\underline{\Delta x}'\underline{G}\underline{\Delta x} \tag{1-11}$$

taking the derivative of Δf with respect to $\underline{\Delta x}$, and equating it to zero, we have

$$\underline{g}' + \underline{\Delta x}'\underline{G} = 0 \tag{1-12}$$

Assuming f is continuous in $\underline{x}$, then $\underline{G}$, the matrix of second partial derivatives with respect to adjustables, is symmetric. Then, from (1-12)

$$\underline{\underline{\Delta x}} = -\ \underline{G}^{-1}\underline{g} \tag{1-13}$$

In this method $\underline{\underline{\Delta x}}$ is called the second order move vector, and the matrix $\underline{G}$ the Hessian matrix.

Since in general f is of higher order than two, a gain λ may be used to find a slightly altered $\underline{\underline{\Delta x}}$ which will minimize f in the direction of $\underline{\underline{\Delta x}}$. The optimal value of λ may be obtained from a one-dimensional search technique. Then

$$\underline{\underline{\Delta x}} = -\lambda \underline{G}^{-1}\underline{g} \tag{1-14}$$

For a non-trivial solution of $\underline{\underline{\Delta x}}$, the correction will yield the maximum value of $|\Delta f|$. In order for Δf to be negative, it is necessary that $\underline{G}$, evaluated at the point $\underline{x}_0$, be positive definite. If the matrix $\underline{G}$ is not positive definite, the quadratic approximation would have concave curvature with respect to $\underline{x}$, and the vector of corrections would lead to a maximum instead of a minimum. In this case the one-dimensional search would set $\lambda = 0$ and one would not be able to move from $\underline{x}_0$ anymore.

Theoretically this algorithm shows a very fast convergence to the minimum of f, in regions close to it, since any function, with continuous partials, is nearly quadratic in the neighborhood of a minimum, if this minimum exists.

When using the digital computer this method presents the disadvantage that the computation of the inverse at each iteration of the Hessian matrix $\underline{G}$ is tedious, especially when the dimension of $\underline{x}$ is very large.

1.4 Fletcher and Powell's method.

This method employs a positive definite matrix $\underline{H}$, which replaces the evaluation of the matrix $\underline{G}$ and its inverse, as needed in the second order move vector method.

Initially the matrix $\underline{H}$ may be set as the identity matrix, and then by an iterative proceedure, $\underline{H}$ is made to converge to $\underline{G}^{-1}$. At each iteration $\underline{H}$ has added a correction which makes it closer to the value of $\underline{G}^{-1}$, which is used to obtain the vector of corrections as in the second order move vector technique.

The one-dimensional search for an optimal gain, is now replaced by a cubical interpolation to find the best gain α, where α is the optimum λ. The vector of corrections for a second order function is therefore expressed as

$$\underline{s}^i = - \underline{H}^i \underline{g}^i \tag{1-15}$$

For a higher order function f the vector of corrections is

$$\underline{\Delta x}^i = -\alpha^i \underline{H}^i \underline{g}^i = \alpha^i \underline{s}^i \tag{1-16}$$

where i denotes the i^{th} iteration. The algorithms for correcting $\underline{H}$ and finding the scalar are given as follows:

$$\underline{H}^{i+1} = \underline{H}^i + \underline{A}^i + \underline{B}^i \tag{1-17}$$

where

$$\underline{A}^i = \frac{\underline{\Delta x}^i (\underline{\Delta x}^i)'}{(\underline{\Delta x}^i)'(\underline{g}^{i+1} - \underline{g}^i)} \tag{1-18}$$

$$\underline{B}^i = \frac{\underline{H}^i(\underline{g}^{i+1} - \underline{g}^i)(\underline{g}^{i+1} - \underline{g}^i)'\underline{H}^i}{(\underline{g}^{i+1} - \underline{g}^i)'\underline{H}^i(\underline{g}^{i+1} - \underline{g}^i)} \tag{1-19}$$

The scalar at the i^{th} iteration is given by

$$\frac{\alpha^i}{\eta} = 1 - \frac{(\underline{h}^i)'\underline{s}^i + w + z}{(\underline{h}^i)'\underline{s}^i - (\underline{g}^i)'\underline{s}^i + 2w} \tag{1-20}$$

where

$$\eta = \min\left(1, \frac{-2(f(\underline{x}_o^i) - f_{min}(\underline{x}))}{(\underline{g}^i)'\underline{s}^i}\right) \tag{1-21}$$

$$z = \frac{3}{\eta}(f(\underline{x}_o^i) - f(\underline{y}^i)) + (\underline{g}^i)'\underline{s}^i + (\underline{h}^i)'\underline{s}^i \tag{1-22}$$

$$w = (z^2 - (\underline{g}^i)'\underline{s}^i(\underline{h}^i)'\underline{s}^i)^{\frac{1}{2}} \tag{1-23}$$

$$\underline{y}^i = \underline{x}_o^i + \eta\underline{s}^i \tag{1-24}$$

and $(\underline{h}^i)'$ is the transpose of the n-vector whose elements are given by $h_i^i = \partial f/\partial y_i^i$.

If $\underline{H}^i$ is correct on the i^{th} iteration, then $\underline{H}^{i+1}$ is also correct. Therefore starting with a correct $\underline{H}$ makes Fletcher and Powell's method converge in only a few steps. Fletcher and Powell's method converges in n steps for an n-order vector $\underline{s}$.

CHAPTER 2

THE POWER PROBLEM

2.1 Definition of the error function.

Given an operating electric power system with a certain number of nodes, and the real and reactive powers at each of them, the problem is to converge from an assumed set of node voltages, to an exact solution. We define an exact solution as the set of node voltages, or vector of voltages, whose corresponding real and reactive powers at each node match the operating powers. This suggests an error definition for a given vector of voltages. Let Pt_j and Qt_j represent the real and reactive powers at node j, respectively, obtained from the operating system. These values can be considered as targets for node j. Now let P_j and Q_j be the real and reactive powers at node j corresponding to an assumed set of node voltages. Then the errors in powers at node j may be defined as

$$PE_j = P_j - Pt_j \tag{2-1}$$

and

$$QE_j = Q_j - Qt_j \tag{2-2}$$

To converge to an exact solution, it is necessary to minimize these errors at every node until they become zero.

In order to apply the minimization techniques already mentioned, we need to define an error function. It is an absolute requirement that the minimum of this function corresponds to zero power errors at every node, and also that its first and second partial derivatives with respect to the voltages, exist at every point. The function selected is

$$f = \sum_{j=1}^{NN} (CP_j^2 x PE_j^2 + CQ_j^2 x QE_j^2)/(Pt_j^2 + Qt_j^2) \tag{2-3}$$

where

NN = number of nodes

CP_j = sensitivity coefficient for real power error at node j. This value varies between zero and unity depending on the accuracy of the measurement of the real power target.

CQ_j = sensitivity coefficient for reactive power error at node j. This value varies between zero and unity depending on the accuracy of the measurement of the reactive power target.

Note that the minimum of this function is zero and corresponds to zero power errors. Also note that the first and second derivatives of the power errors with respect to the voltages, correspond to the derivatives of the real and reactive powers with respect to the voltages, which can be determined analytically and always exist.

The voltages here can be separated in real and imaginary part, or in magnitude and angle. This last approach is used in the minimization techniques, where half of the elements of the n-vector of adjustables $\underline{x}$ are voltage magnitudes only, and the other half voltage angles.

It is also very important to note that a reference angle is needed at one of the nodes, that is, if one has NN nodes, only (NN-1) angles are to be corrected, while one is kept constant throughout all the minimization process.

2.2 A first order approach to the power problem. Newton's method.

This is actually a modification of Newton's method used to solve the power problem.

The corrections to voltage magnitudes and angles are found by a linear approximation, that gives rise to voltages whose powers match the targets according to this approximation. The basic relation is

$$\begin{bmatrix} \underline{QE} \\ \hline \underline{PE} \end{bmatrix} = - \left[\begin{array}{c|c} (\partial Q/\partial V) & (\partial Q/\partial A) \\ \hline (\partial P/\partial V) & (\partial P/\partial A) \end{array}\right] \begin{bmatrix} \underline{\Delta V} \\ \hline \underline{\Delta A} \end{bmatrix} \qquad (2\text{-}4)$$

where $\underline{\Delta V}$ is the NN-vector of voltage magnitude corrections; $\underline{\Delta A}$ is the (NN-1)-vector of angle corrections; $(\partial Q/\partial V)$ is the NNxNN matrix of partial derivatives of reactive power with respect to voltage magnitudes; $(\partial Q/\partial A)$ is the NNx(NN-1) matrix of partial derivatives of reactive power with respect to angles; and, similarly, $(\partial P/\partial V)$ is (NN-1)xNN; $(\partial P/\partial A)$ is (NN-1)x(NN-1). These matrices are evaluated at each iteration. $\underline{QE}$ and $\underline{PE}$ are the NN-vector and (NN-1)-vector of reactive and real power errors respectively. The expression for the corrections is as follows:

$$\begin{bmatrix} \underline{\Delta V} \\ \hline \underline{\Delta A} \end{bmatrix} = - \left[\begin{array}{c|c} (\partial Q/\partial V) & (\partial Q/\partial A) \\ \hline (\partial P/\partial V) & (\partial P/\partial A) \end{array}\right]^{-1} \begin{bmatrix} \underline{QE} \\ \hline \underline{PE} \end{bmatrix} \qquad (2\text{-}5)$$

If a solution for zero power error exists, and partial derivatives are continuous in the neighborhood of this solution, which is the case for the power problem, then the method should converge. If a solution for small $\underline{QE}$ and $\underline{PE}$ is the only one that exists, then the method works well at the first few iterations, but become unstable in the neighborhood of the best solution.

Best results are achieved by adjusting angles alone at odd iterations, i.e.

$$\underline{\Delta A} = - (\partial P/\partial A)^{-1}\underline{PE} \qquad (2\text{-}6)$$

and adjusting the entire state vector of voltage magnitudes and angles, on even iterations, except the angle at the reference node. For even better results the one-dimensional search technique is also used here.

CHAPTER 3

THE DIGITAL COMPUTER APPLICATION

3.1 Introduction.

The digital computer was utilized to evaluate the efficiency of second order minimization techniques when compared to first order approaches, such as steepest descent and Newton's method, to obtain the solution of the power problem. The CDC 6400 machine was used for this purpose.

The data for the power system was an eight node model obtained from an actual power system in Australia. The available data consisted of the admittance matrix, and the measured real and reactive powers at each node. It is worthwhile to notice that any measurement implies some error, so it was expected that the target powers imply a solution whose minimum error is different from zero. In view of this, the convergence criterion was also tested using targets with an exact solution, that is, a solution which yields a minimum error equal to zero. These targets were obtained proceeding backwards, that is, a solution was assumed and then the real and reactive powers were obtained and used as targets.

The initial estimate for adjustables in each method was set with voltage magnitudes equal to 1.0 per unit value, and angles being equal to -0.3 radians. The angle at node 8 was held constant, i.e., chosen as the reference angle.

3.2 Programs tested.

Computer programs were written in order to test the following techniques:

(a) Newton's method only

(b) Newton's method for the first two iterations and then a modification of Fletcher and Powell's method. This modification consists of computing the exact second partial derivatives of the error function with respect to the adjustables, i.e. the matrix $\underline{G}$, invert the matrix, and use it for the first iteration with Fletcher and Powell's method, instead of using the unit matrix as the initial $\underline{H}$ matrix.

(c) Newton's method for the first two iterations and then the second order move vector technique.

(d) Newton's method for the first two iterations and then Fletcher and Powell's method.

(e) Fletcher and Powell's method only.

(f) Steepest descent method only.

(g) Second order move vector method only.

3.3 Results of the different programs.

The results of the different programs mentioned above, are shown graphically in figures 3-1 and 3-2, where the plots show per unit RMS error against computer time in the CDC 6400 machine. This per unit RMS error, called here EPU, is calculated at each iteration from the error defined in equation (2-3), and is given by

$$EPU = (f/2x\text{No. of nodes})^{\frac{1}{2}} \qquad (3\text{-}1)$$

The case of an exact solution is considered as a case of well measured target powers, while the actual case, as a case of not well measured target powers.

Steepest descent method is not shown in the plots because its convergence was too slow to be considered together with the other methods. The second order move vector technique operating alone is not shown either, since convergence was not achieved because the initial estimate is in a region where the matrix $\underline{G}$ is not positive definite. Only the use of the second order move vector method together with Newton's method is shown in the plots.

In order of computational speed, for a well measured target set, the methods are (a), (b), (c), (d), and (e). For a poorly measured target set, corresponding to actual system data, the methods in order of computational speed are (b), (c), (d), (e), and (a). Newton's method was placed last because it did not calculate the absolute minimum error.

Figure 3-3 is the one-line circuit diagram of the ten-node high voltage transmission system for New South Wales, Snowny Mountain Scheme, and Victoria.

Tables I, II, and III show the generator constants, line characteristics, and peak power flows respectively for the year 1974.

GENERATOR RATINGS

TABLE I.

Node	Inertia MW-SEC.	Rated MVA	Rated MW	X_q	X_d	X_q'	X_d'
				Percent of 100 MVA base.			
1	5,710	2350	2000	10.6	10.6	10.6	1.13
2	8,710	2932	2605	7.3	7.3	7.3	0.93
3	2,640	500	450	42.8	42.8	42.8	4.6
4	7,110	1580	1500	5.5	9.7	5.5	2.42
5	8,090	2247	2100	2.7	4.7	2.7	1.26
6	10,530	3000	2710	6.9	6.9	6.9	0.81

TABLE II.

Line Characteristics

Nodes	Circuits	Line Charging Capacitance		Impedance	
		MVARS	per unit admittance 100 MVA base	Percent of 100 MVA base	
				R	X
1 2	1	34	.34	0.38	4.15
1 3	2	170	1.7	0.13	1.52
2 3	4	240	2.4	0.07	0.72
3 7	4	225	2.25	0.2	1.73
4 7	3	120	1.2	0.1	1.0
4 5	3	40	.4	0.11	0.85
5 10	2	95	.95	0.21	1.37
10 8	2	194	1.94	0.46	5.08
6 8	6	60	.6	0.21	1.48
6 9	1	0	0	0	0.88
9 8	3	660	6.6	0.07	1.69

TABLE III.

1974 Peak Load Power Flows

Node		Power Flow MW	MVAR
1	Liddell	1920	200
2	Vales Pt.	2300	320
3	Sydney	430	95
3		-5245	0
4	Lower Tumut	1500	175
5	Murray	2100	250
6	Latrobe	2600	870
6		-450	-260
7	Yass	-490	0
8	Melbourne	-3150	-1150
9	Hazelwood	0	
10	Dederang	0	

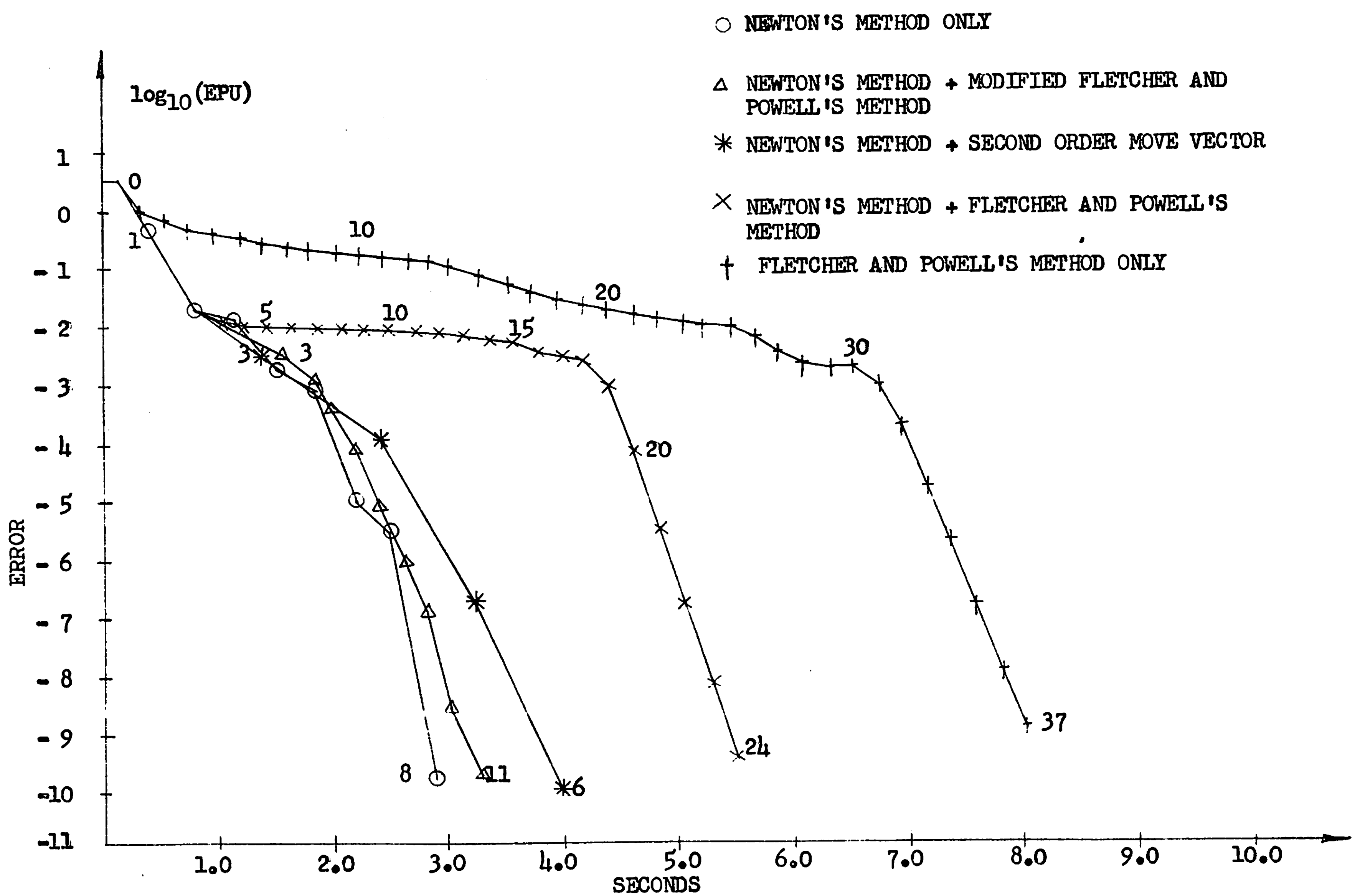


Fig. 3-1. Convergence to a well measured target set.

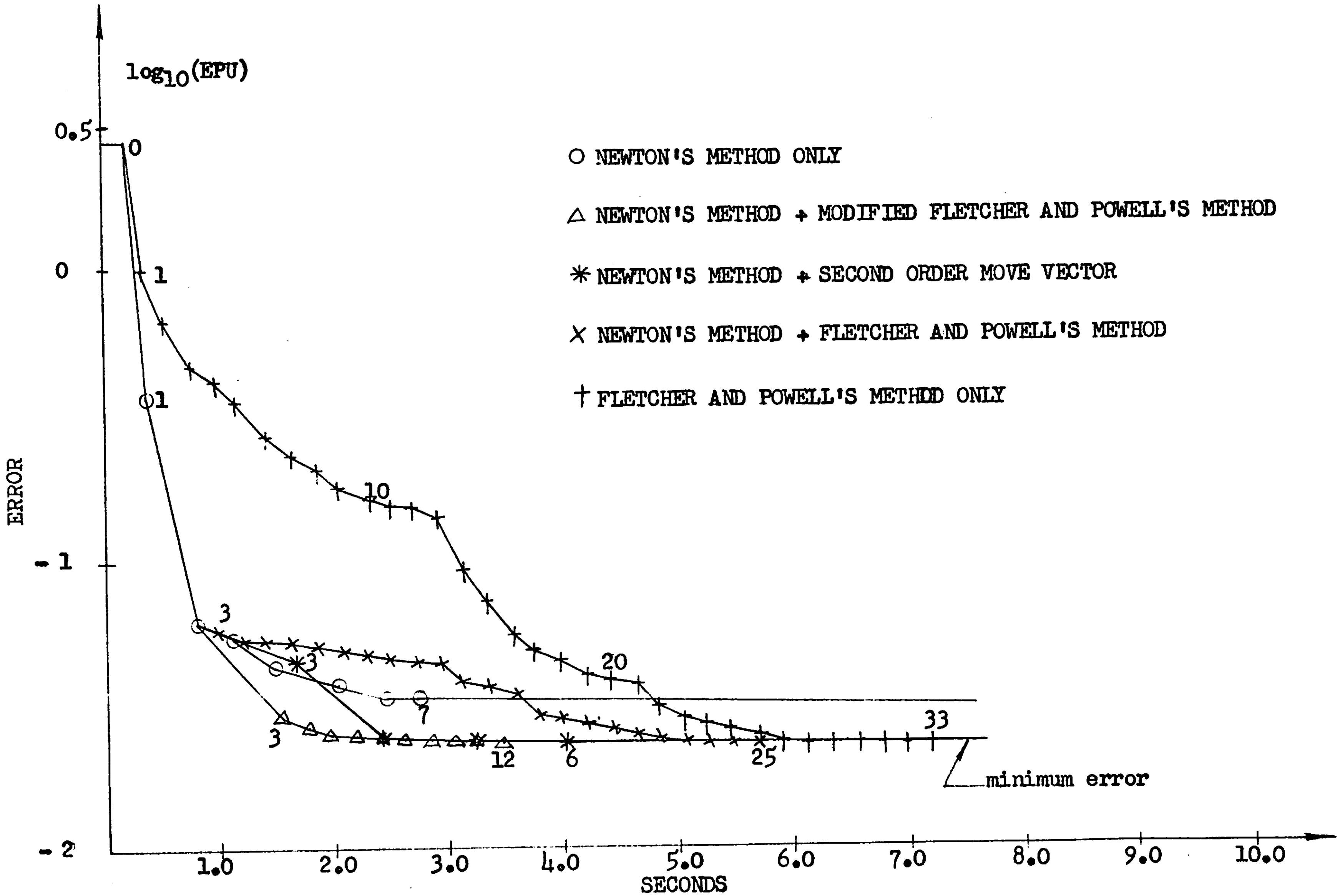


Fig. 3-2. Convergence to a not well measured target set.

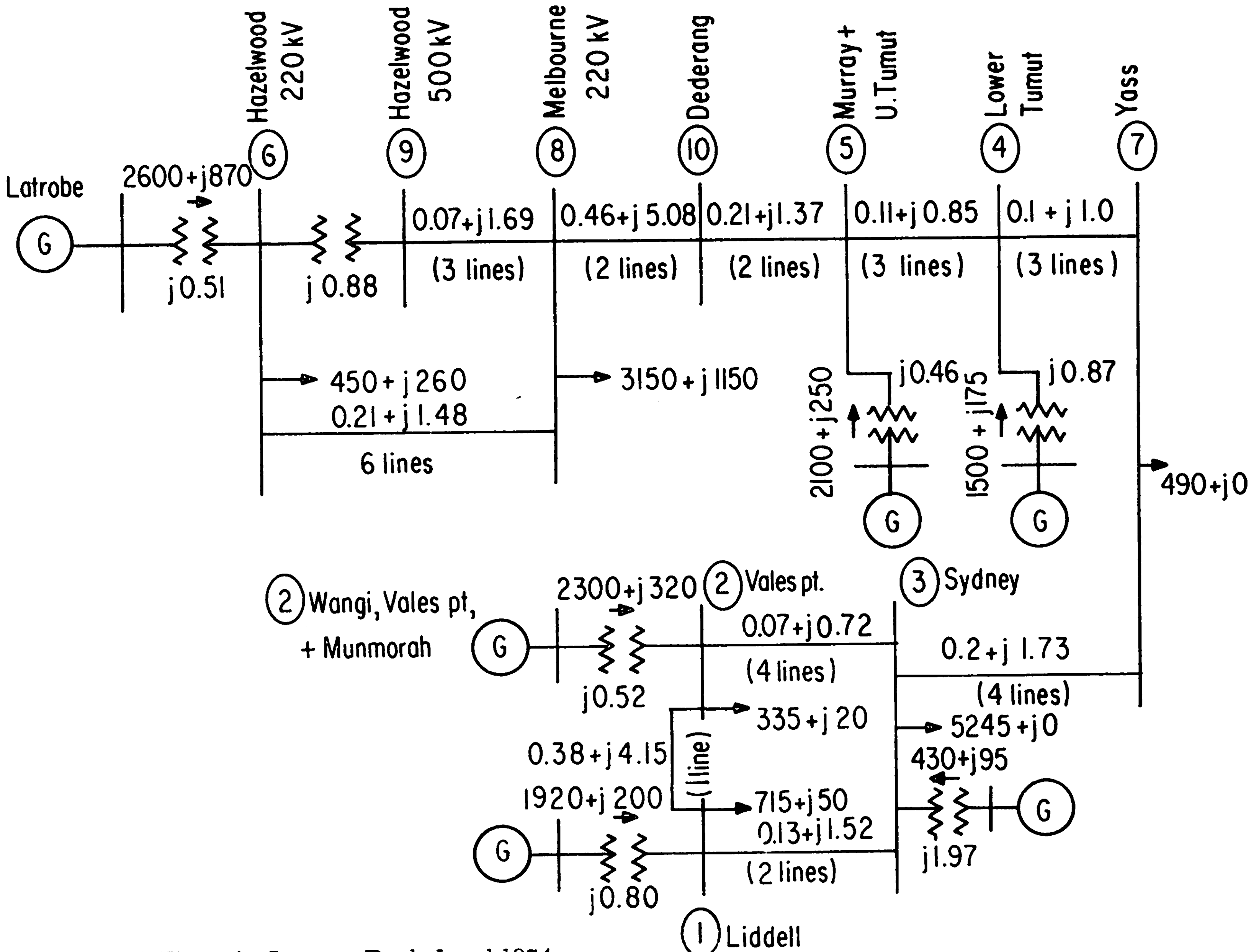


Fig. 3-3. New South Wales and Victoria System Peak Load 1974.
Units: Impedances are Percent of 100 MVA Base.
Inertia Constant K in MW-SEC.
Generation and Loads in MW and MVAR.

CHAPTER 4
FINAL COMMENTS

4.1 Conclusions.

From the plots it is seen that Newton's method achieves the fastest convergence when an exact solution for the system exists. In the real case this does not happen, hence Newton's method in combination with the modified Fletcher and Powell's method, is the fastest technique to use. In the real case Newton's method tries to go unstable after seven iterations, but the one-dimensional search for the optimal gain sets this one equal to zero. So the error remains unchanged after the seventh iteration.

Notice that all the second order techniques converge to exactly the same minimum error, and stay there because the corrections become zero (in computer terms) when the gradient is zero, i.e., at the bottom of the error surface in the space of adjustables.

The computer programs used are independent of the number of nodes in the system, however, when a system with large number of nodes is tested, some of the computations, especially those involving matrix operations, may become inefficient.

Another problem that arises when the power problem involves a very large system, is the limitation in core storage in the computer. This can be solved with an intelligent use of the magnetic tape available in almost any digital computer facility.

With regard to the power problem itself, the fastest program may be used to obtain the steady state of the operating system, either in a pretransient situation, or in the posttransient if a fault or an intentional cut out accurs. In the posttransient case, a method might be developed to update all the matrices and vectors, as left from the last iteration in the pretransient, without having to calculate them explicitly.

Sparce matrix techniques were not used for the simple case reported here, but should be used for the implementation of these methods for analysis and control of a large power system.

BIBLIOGRAPHY ON ALGORITHMS

(1) R. FLETCHER and M.J.D. POWELL, "A rapidly convergent descent method for minimization", Computer Journal, vol. 6, pp. 163-168, 1963.

(2) G.E. MAUERSBERGER, "Power system realizable state", Research Project, Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, California, March 8, 1968.

(3) R.E. BELLMAN and S.E. DREYFUS, "Applied dynamic programming", Princeton University Press, Princeton, New Jersey, 1962.

(4) A.S. HOUSEHOLDER, "Principles of numerical analysis", McGraw - Hill, 1953.

BIBLIOGRAPHY ON SPARSE MATRIX METHODS

(5) F. GUSTAVSON, W. LINIGER, and R. WILLOUGHBY, "Symbolic generation of an optimal Crout algorithm for sparse systems of linear equations", Research Paper, RC-1852, IBM Watson Research Center, Yorktown Heights, New York, 1967.

(6) R.P. TEWARSON, "The product form of inverses of sparse matrices and graph theory", SIAM Review, pp. 91-99, 1967.

(7) N. SATO and W.F. TINNEY, "Techniques for exploiting the sparsity of the network admittance matrix", IEEE Trans. on Power Apparatus and Systems, pp. 944-950, August 2, 1963.

(8) A.M. HERSHDORFER, and G.M. STURMAN, "On the efficient inversion of large structured matrices", Proc. Engineering Mechanics Division Speciality Conference, ASCE, Washington, D.C., October 1966.

BIBLIOGRAPHY ON CONTROL METHODS

(1) Concordia, C., "Steady-State Stability of Synchronous Machines as Affected by Angle-Regulator Characteristics", Trans. AIEE, Vol. 67, Part I, 1948, pp. 687-690, Digest in Electrical Engineering, Vol. 67, No. 6, June 1948, p. 570.

(2) Smith, Otto J.M., "Synchronous - Flux Generator", Electrical Engineering, July 1958, pp. 605-610.

(3) Turvey, W. L., Constant-Frequency Alternating-Current Generators, British Patent and U. S. Patent No. 2,829,333, April 1, 1958.

(4) Johnson, L. J., Frequency Control Apparatus for Alternators, U.S., Patent No. 2,854,617, September 30, 1958.

(5) Gibson, D. K., Frequency Stabilization System, U. S. Patent No. 2,886,766, May 12, 1959.

(6) Chirgwin, K. M. and Stratton, L. J., "Variable-Speed Constant-Frequency Generator System for Aircraft", AIEE Transactions, Vol. 78, Part II, Applications and Industry, No. 45, Nov. 1959, pp. 304-310.

(7) Jones, Graham Allan, "Transient Stability of a Synchronous Generator Under Conditions of Bang-Bang Excitation Scheduling Following Load Rejection", M.S. Thesis, University of California, Berkeley, January 1963.

(8) Mottershead, Allen, "Machine Suboptimal Phase-Plane Excitation Control of Rotor Transients," M.S. Thesis, University of California, Berkeley, September 1, 1964.

(9) Smith, Otto J. M., "Synchronous Machine Stability Enhancement with State Space Switching," Anais do Primeiro Congress Nacional de Engenharia Electronica, Instituto Tecnologico de Aeronautica, Sao Jose dos Campos, Brasil, January 4, 1965.

(10) Jones, Graham Allan, "Transient Stability of a Synchronous Generator Under Conditions of Bang-Bang Excitation Scheduling," IEEE Trans. on Power Apparatus and Systems, Vol. PAS-84, No. 2, pp. 114-121, February 1965.

(11) Smith, Otto J.M., Discussion on above, IEEE Trans. on Power Apparatus and Systems, Vol. PAS-84, No. 2, p. 120, February 1965.

(12) Smith, Otto J. M., "Optimal Transient Removal in a Power System," IEEE Trans. on Power Apparatus and Systems, Vol. PAS-84, No. 5, pp. 361-374, May 1965, Discussion, PAS-84, No. 6, pp. 528-531, June 1965.

(13) Parikh, Praduman J., "Forced Excitation for Synchronous Machine Stabilizing", M.S. Thesis, University of California, Berkeley, June, 1965.

(14) Smith, Otto J. M., Discussion on "Effect of Turbine-Generator Representation in System Stability Studies", by Tokay and Bolger, IEEE Trans. on Power Apparatus and Systems, Vol. PAS-84, No. 10, p. 941, October 1965.

(15) Smith, Otto J. M., Discussion on "Inverter Rotor Drive of an Induction Motor" by M.S. Erlicki, IEEE Trans. on Power Apparatus and Systems, Vol. PAS-84, No. 11, p. 1016, November 1965.

(16) Hiles, Leonard A., "Suboptimal Excitation Control of Synchronous Machines," M.S. Thesis, University of California, Berkeley, January 1966.

(17) Smith, Otto J. M., "Protective Devices should Guard System Rather than its Components," IEEE Spectrum, Vol. 3, No. 5, pp. 89-90, May 1966.

(18) Fairfield, Roger L., "A Posicast Transient Quencher," M.S. Thesis, University of California, Berkeley, 1966.

(19) Smith, Otto J. M., "System, Apparatus and Method for Improving Stability of Synchronous Machines," U. S. Patent 3,388,305, June 11, 1968.

(20) Smith, Otto J. M., "Power System Transient Control by Capacitor Switching," IEEE Winter Power Meeting Paper No. 68-TP-115-PWR, January, 1968.

(21) de Mello, F. P., C. Concordia, "Concepts of Synchronous Machine Stability as Affected by Excitation Control," IEEE Winter Power Meeting Paper No. 68-TP-129-PWR.

APPENDIX

COMPUTER PROGRAMMING

```
C
C
      A.1  GLOSSARY OF TERMS.
C
C     A          ELEMENT OF THE MATRIX OF SECOND PARTIAL DERIVATIVES OF
C                THE ERROR FUNCTION WITH RESPECT TO ADJUSTABLES. IN SUBROU
C                TINE NEWTON THIS IS AN ELEMENT OF THE JACOBIAN MATRIX. IN
C                SUBROUTINE ASSEMAD IT IS AN ALEMENT OF THE ADMITTANCE MA-
C                TRIX, WITH REAL PART AR AND IMAGINARY PART AI.
C     AC         ELEMENT OF ONE OF THE TWO MATRICES OF CORRECTIONS IN SUB-
C                ROUTINES MODFPOW AND FPOWELL.
C     AL         GAIN GIVEN AS AN INPUT TO SUBROUTINE FUNCT.
C     ALAMB      GAIN TO BE USED WITH THE VECTOR OF CORRECTIONS IN NNEWT.
C     ALPHA      GAIN TO BE USED WITH THE VECTOR OF CORRECTIONS AS GIVEN
C                FROM SUBROUTINES MODFPOW AND FPOWELL.
C     ANGDIFF    DIFFERENCE IN ANGLE BETWEEN TWO NODES.
C     ANGMACH    GENERATOR ROTOR ANGLE.
C     ANGTER     TERMINAL VOLTAGE ANGLE.
C     AP         STORAGE OF SECOND PARTIAL DERIVATIVE OF REAL POWER WITH
C                RESPECT TO ADJUSTABLES.
C     AQ         STORAGE OF SECOND PARTIAL DERIVATIVE OF REACTIVE POWER
C                WITH RESPECT TO ADJUSTABLES.
C     AVEANG     AVERAGE ROTOR ANGLE OF GENERATORS.
C     BB         SCRATCH VARIABLE.
C     BC         ELEMENT OF ONE OF THE TWO MATRICES OF CORRECTIONS IN SUB-
C                ROUTINES MODFPOW AND FPOWELL.
C     CP         SENSITIVITY COEFFICIENT FOR REAL POWER ERROR.
C     CQ         SENSITIVITY COEFFICIENT FOR REACTIVE POWER ERROR.
C     D          COMPOSITE CORRECTION FOR ADJUSTABLE.
C     DEL        SCRATCH VARIABLE.
C     DELX       NEGATIVE OF ELEMENT OF THE VECTOR OF CORRECTIONS IN MAIN
C                PROGRAM. ELEMENT OF THE GRADIENT IN SUBROUTINE GRAD1. SE-
C                COND ORDER MOVE VECTOR IN SUBROUTINE METHOD.
C     DELXO      SCRATCH VARIABLE.
C     DEPU       DIFFERENCE IN RMS PER UNIT ERROR IN TWO ITERATIONS.
C     DPDA       PARTIAL DERIVATIVE OF REAL POWER WITH RESPECT TO ANGLE.
C     DPDV       PARTIAL DERIVATIVE OF REAL POWER WITH RESPECT TO VOLTAGE.
C     DQDA       PARTIAL DERIVATIVE OF REACTIVE POWER WITH RESPECT TO AN-
C                GLE.
C     DQDV       PARTIAL DERIVATIVE OF REACTIVE POWER WITH RESPECT TO VOL-
C                TAGE.
C     ENA        TWICE THE NUMBER OF REGULATED VOLTAGE NODES.
C     ENORMSQ    ERROR FUNCTION USED BY THE MINIMIZATION TECHNIQUES.
C     EPSI       TOLERANCE FOR MINIMUM OF ERROR FUNCTION.
C     EPU        VALUE OF ERROR FUNCTION IN RMS PER UNIT.
C     EPUS       SCRATCH VARIABLE TO STORE VALUE OF EPU.
C     ETA        PRELIMINARY GAIN IN SUBROUTINES MODFPOW AND FPOWELL.
C     F          SENDING-END VOLTAGES RECIPROCAL ADMITTANCE MATRIX L BY M.
C     FAL        VALUE OF ERROR FUNCTION ASSOCIATED WITH GAIN AL IN SUBROU
C                TINE FUNCT.
C     G          RECEIVING END VOLTAGES RECIPROCAL ADMITTANCE MATRIX L BY
C                M. SCRATCH VARIABLE IN MODFPOW AND FPOWELL.
C     H          F-G
C     ICOUNT     COUNTER OF NUMBER OF ITERATIONS IN MAIN PROGRAM.
C     ICOUNTF    LOCAL COUNTER FOR SUBROUTINES MODFPOW AND FPOWELL.
```

```
C       IE         COUNTER OF THE NUMBER OF TIMES EPU HAS NOT CHANGED MORE
C                  THAN E-04 .
C       INDEX      COUNTER OF NUMBER OF ERROR FUNCTIONS.
C       IRROR      DETECTOR FOR SINGULARITY OF MATRICES.
C       JCOUNT     COUNTER OF NUMBER OF CARDS READ AS MACHINE CHARACTERIS-
C                  TICS.
C       K          SCRATCH VARIABLE TO SKIP THE USE OF SUBROUTINE ESPD.
C       L          NUMBER OF MACHINES WITH VOLTAGE REGULATORS. MAXIMUM NUM-
C                  BER OF ITERATIONS IN SUBROUTINE ODS.
C       LC         NUMBER OF CIRCUIT LOST.
C       LN         NUMBER OF LINE LOST.
C       M          MAXIMUM NUMBER OF ITERATIONS IN MAIN PROGRAM. NUMBER OF
C                  MACHINES WITH VOLTAGE REGULATORS IN SUBROUTINE ASSEMAD.
C       MD         NUMBER OF REGULATED VOLTAGE NODES.
C       N          DIMENSION OF VECTOR OF ADJUSTABLES.
C       NF         NUMBER OF ERROR FUNCTIONS TO USE.
C       NM         NUMBER OF GENERATORS.
C       NN         SCRATCH VARIABLE.
C       NODES      TOTAL NUMBER OF NODES.
C
C       POWER BASE IS 100 MEGAWATTS.
C
C       PE         ERROR IN REAL POWER.
C       POWER      COMPUTED REAL POWER AT EACH NODE.
C       POWERS     COMPUTED REAL POWER AT EACH NODE.
C       PSPIN      MAXIMUM REAL POWER AVAILABLE FROM EACH GENERATOR.
C       PTAR       REAL POWER TARGET AT EACH NODE.
C       QE         ERROR IN REACTIVE POWER.
C       QTAR       REACTIVE POWER TARGET AT EACH NODE.
C       R          RECIPROCAL ADMITTANCE REGULATED VOLTAGES.
C       RATEFQ     RATE OF CHANGE OF FREQUENCY IN PERCENT PER SECOND.
C       RNI        IMAGINARY PART OF ADMITTANCE BETWEEN TWO REGULATED VOLTA-
C                  GE NODES.
C       RNR        REAL PART OF ADMITTANCE BETWEEN TWO REGULATED VOLTAGE NO-
C                  DES.
C       ROIN       ROTOR INERTIA. GIVEN IN 100 MEGAWATT-SECOND STORED ENERGY.
C       RZ         REAL PART OF INTERNAL IMPEDANCE.
C       S          ELEMENT OF VECTOR OF CORRECTIONS IN SUBROUTINES MODFPOW
C                  AND FPOWELL.
C       SUMIN      SUM OF INERTIAS OF MACHINES.
C       SUMLOSS    SUM OF REAL POWER LOSSES IN THE SYSTEM.
C       SUMP       SUM OF REAL POWER TARGETS.
C       SUMQ       SUM OF REACTIVE POWER TARGETS.
C       SUMVA      SUM OF VOLTAMPERE TARGETS.
C       TOLER      TOLERANCE FOR SMALLEST INTERVAL IN SUBROUTINE ODS.
C       TOTALP     SUM OF COMPUTED REAL POWERS.
C       TOTALPI    SUM OF REAL POWER INPUTS TO THE SYSTEM.
C       TOTALPO    SUM OF REAL POWER OUTPUTS FROM THE SYSTEM.
C       TOTALQ     SUM OF COMPUTED REACTIVE POWERS.
C       TOTALVA    SUM OF COMPUTED VOLTAMPERES.
C       UMI        IMAGINARY PART OF THE CURRENT AT EACH NODE.
C       UMR        REAL PART OF THE CURRENT AT EACH NODE.
C       VAMPS      COMPUTED VOLTAMPERES AT EACH NODE.
C       VAR        COMPUTED REACTIVE POWER AT EACH NODE.
C       VARS       COMPUTED REACTIVE POWER AT EACH NODE.
```

```
C      VATAR      VOLTAMPERE TARGET AT EACH NODE.
C      VI         IMAGINARY PART OF TERMINAL VOLTAGE.
C      VM         MAGNITUDE OF MACHINE VOLTAGE.
C      VMI        IMAGINARY PART OF MACHINE VOLTAGE.
C      VMR        REAL PART OF MACHINE VOLTAGE.
C      VR         REAL PART OF TERMINAL VOLTAGE.
C      VREF       MAGNITUDE OF TERMINAL VOLTAGE.
C      X          ELEMENT OF VECTOR OF ADJUSTABLES. REACTANCE OF TRANSMI-
C                 SSION LINES IN SUBROUTINE ASSEMAD.
C      XMAX       OPTIMAL GAIN GIVEN BY SUBROUTINE ODS.
C      XZ         IMAGINARY PART OF INTERNAL IMPEDANCE.
C      YMAX       MINIMUM VALUE OF ERROR FUNCTION ASSOCIATED WITH THE OPTI-
C                 MAL GAIN IN SUBROUTINE ODS.
C      ZMI        SYNCHRONOUS REACTANCE OF GENERATOR.
C      ZMR        RESISTANCE OF GENERATOR.
C      ZI         INVERSE ADMITTANCE MATRIX, IMAGINARY PART.
C      ZR         INVERSE ADMITTANCE MATRIX, REAL PART.
C      ZZ         ABSOLUTE VALUE OF INTERNAL IMPEDANCE.
C
C      ORDER OF NUMBERING NODES.
C
C      0 GROUND OR REFERENCE NODE
C      1 LIDDELL EXCITATION                1 LIDDELL TERMINALS
C      2 WANGI EXCITATION                  2 WANGI TERMINALS
C      3 SYDNEY EXCITATION                 3 SYDNEY TERMINALS
C      4 LOWER TUMUT EXCITATION            4 LOWER TUMUT TERMINALS
C      5 MURRAY EXCITATION                 5 MURRAY TERMINALS
C      6 LATROBE EXCITATION                6 LATROBE TERMINALS
C                                          7 YASS
C                                          8 MELBOURNE
C                                          9 HAZELWOOD
C                                         10 DEDERANG
C      VALUES FOR 1974 PEAK LOAD
```

Subroutine	Function
MODFPOW	Computes vector of corrections using a modification to Fletcher and Powell's method. The modification consists of using the matrix of exact second partial derivatives as the Hessian matrix for the first iteration, instead of the unit matrix. The following iterations are performed according to the original method. The optimal gain is also obtained here using cubical interpolation.
MTINV	Inverts a real matrix and the solution to a system of linear equations can also be obtained.
MTINV1	Uses common blocks in MTINV for particular purposes in connection with the main program.
MTMPB	Multiplies two square matrices.
NEWTON	Computes vector of corrections using Newton's method applied to power systems. At odd iterations only corrections for voltage angles are obtained. At even iterations the entire state vector is corrected.
NNEWT	Main program. It obtains the operating state of a power system given an initial estimated state using error function minimization. The program employs the following minimization techniques:

Subroutine	Function
	(a) NNEWT1. Newton's method only. (b) NNEWT2. Newton's method for the first two iterations and modified Fletcher and Powell's method for the following iterations. (c) NNEWT3. Newton's method for the first two iterations and the second order move vector technique for the following iterations. (d) NNEWT4. Newton's method for the first two iterations and Fletcher and Powell's method for the following iterations. (e) NNEWT5. Fletcher and Powell's method only. (f) NNEWT6. Steepest descent method only. (g) NNEWT7. Second order move vector only.
ODS	Performs a one dimensional search along the curve error vs. gain in order to obtain the optimal gain to be used together with the vector of corrections. The technique employed is the golden mean, derived from the Fibonacci numbers.
REST	Computes the value of the error function for a given n-vector of adjustables $\underline{x}$.

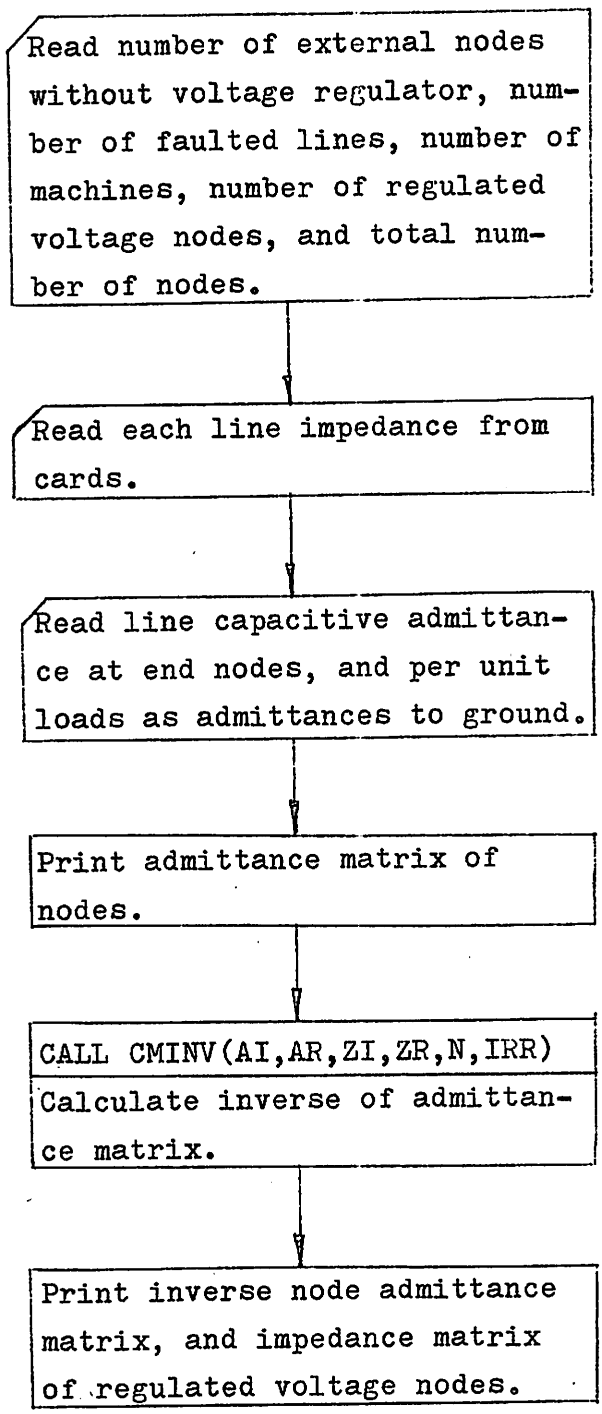
A.3 Computer time for selected subroutines on CDC 6400.

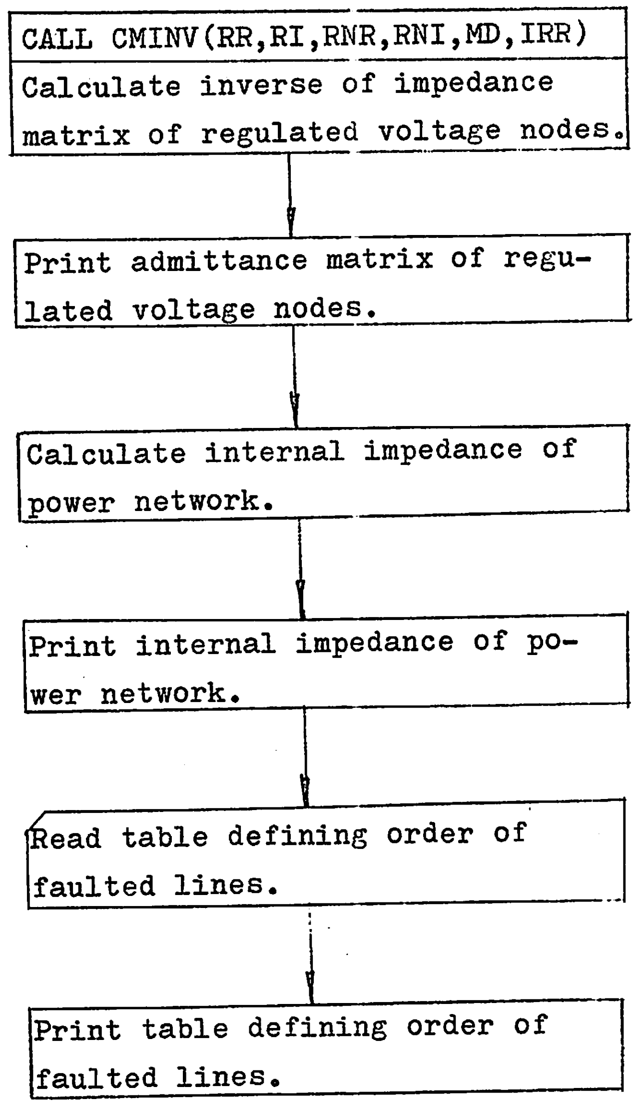
Subroutine	Operation	Time in seconds
ASSEMAD	Obtention of 8 by 8 complex admittance matrix of regulated voltage nodes, including printings.	0.840
CMINV	Inversion of 10 by 10 complex matrix.	0.206
CMINV	Inversion of 8 by 8 complex matrix.	0.114
ESPD	Computation of 15 by 15 matrix of second partial derivatives.	0.466
FPOWELL	Calculation of 15 corrections and optimal gain.	0.124
GRAD1	Computation of 15 first patial derivatives.	0.082
METHOD	Calculation of 15 corrections.	0.006
MODFPOW	Calculation of 15 corrections in first iteration.	0.640
MODFPOW	Calculation of 15 corrections in subsequent iterations.	0.124

Subroutine	Operation	Time in seconds
MTINV and MTINV1	Inversion of 15 by 15 matrix.	0.176
MTINV and MTINV1	Inversion of 7 by 7 matrix.	0.022
NEWTON	Computation of 7 corrections.	0.030
NEWTON	Computation of 15 corrections.	0.186
ODS and INTER FUNCT	Calculation of optimal gain for vector of 15 corrections. Tolerance= 10^{-3}.	0.182
REST	Computation of value of error function for 16 variables.	0.009

A.4 Flow diagrams and listing of programs.

SUBROUTINE ASSEMAD(RNR,RNI)




CALL CMINV(RR,RI,RNR,RNI,MD,IRR)
Calculate inverse of impedance matrix of regulated voltage nodes.
Print admittance matrix of regulated voltage nodes.
Calculate internal impedance of power network.
Print internal impedance of power network.
Read table defining order of faulted lines.
Print table defining order of faulted lines.

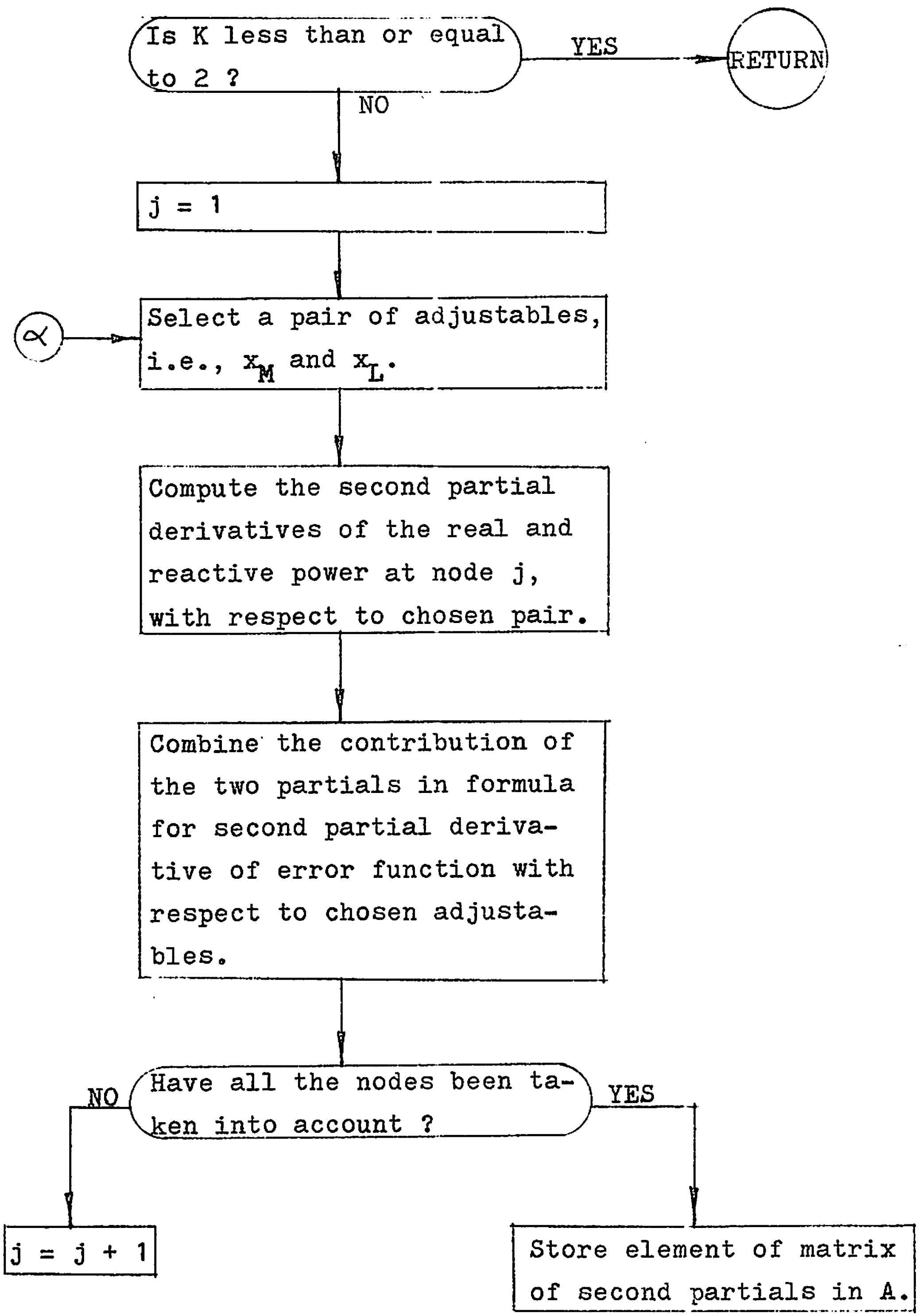
```
      SUBROUTINE ASSEMAD(RNR,RNI)
C     ************************************************************************
C     *THIS SUBROUTINE READS IMPEDANCES OF TRANSMISSION LINES AND COMPU- *
C     *TES THE IMPEDANCE AND ADMITTANCE MATRICES OF THE POWER SYSTEM. IT *
C     *IS USED BY THE MAIN PROGRAM TO OBTAIN THE ADMITTANCE MATRIX OF RE-*
C     *GULATED VOLTAGE NODES.                                            *
C     ************************************************************************
      DIMENSION RR(8,8),RI(8,8),RNR(8,8),RNI(8,8),RCR(8,8),RCI(8,8)
      DIMENSION JF(8),JG(8)
      DIMENSION AR(10,10),AI(10,10),ZR(10,10),ZI(10,10)
      DIMENSION CZR(8,8),CZI(8,8)
      DIMENSION ZCR(8),ZCI(8),ZCZ(8)
C     D     NUMBER OF EXTERNAL NODES WITHOUT VOLTAGE REGULATORS
C     L = NUMBER OF FAULTED LINES
C     M = NUMBER OF MACHINES
C     N IS NUMBER OF NODES
C     MD = M+D
C     CAUTION, DO NOT CHANGE VALUES OF L , M, AND NODES IN PROGRAM
C     THESE ARE RESERVED FOR CONSTANTS
      READ 302,D,L,M,MD,NODES
302   FORMAT (5I5)
      N = NODES
      NS = NODES*NODES
C     CLEARING TO ZERO, HOUSEKEEPING
      DO 2 K = 1,NS
      AR(K) = 0
      AI(K) = 0
      ZR(K) = 0
      ZI(K)=0
2     CONTINUE
C     READ EACH LINE IMPEDANCE FROM CARDS
52    READ 55,J,K,R,X,KK
55    FORMAT (2I5,2F15.6,35X,I5)
      DZ= R*R + X*X
      AR(J,K) = AR(J,K) -R/DZ
      AI(J,K) = AI(J,K) + X/DZ
      IF(KK)87,52,87
C     READ LINE CAPACITIVE ADMITTANCE AT END NODES, B IS POSITIVE
C     LOADS NEAR GENERATORS ARE REPRESENTED BY ADMITTANCES    K=1+NODES
C     READ PERUNIT LOADS AS ADMITTANCES TO GROUND FROM CARDS.  NEG B IS L
87    READ 55,J,K,A,B,KK
      AR(J,J)=AR(J,J)+A
      AI(J,J)=AI(J,J)+B
      IF(KK)88,87,88
88    CONTINUE
C     ADMITTANCE MATRIX WILL BE FILLED OUT BY THE FOLLOWING
      NN = NODES - 1
      DO 101 I = 1, NN
      II = I + 1
      DO 101 J = II, NODES
      AR(J,I) = AR(I,J)
      AI(J,I) = AI(I,J)
      AR(I,I) = AR(I,I) - AR(I,J)
      AI(I,I) = AI(I,I) - AI(I,J)
      AR(J,J) = AR(J,J) - AR(I,J)
```

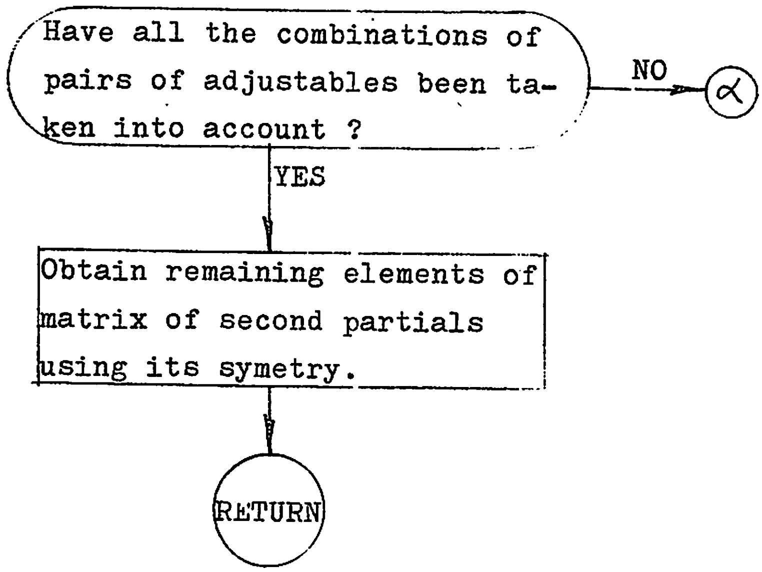
```
101   AI(J,J)=AI(J,J)-AI(I,J)
      PRINT 120
120   FORMAT(//20X,27H ADMITTANCE MATRIX OF NODES   /)
      DO  124  KA=1, NODES, 4
      IF (KA+3-NODES)122,121,121
121   KB=NODES
      GO TO 123
122   KB=KA+3
123   PRINT 313,(JCH,JCH=KA,KB)
313   FORMAT(/4I30/)
      DO 124  J=1, NODES
124   PRINT 316, J, (AR(J,K), AI(J,K), K=KA,KB)
316   FORMAT(I5,4(F17.2,F13.2))
      PRINT 330
330   FORMAT (//)
C     CALCULATION OF INVERSE ADMITTANCE MATRIX
      DO 126 J=1,NS
126   AI(J)=-AI(J)
      N=NODES
C     SET DIMENSIONS IN SUBROUTINES TO N  THIS NOT AUTOMATIC
      CALL CMINV(AI,AR,ZI,ZR,N,IRR)
      DO 131 J=1,NS
131   ZR(J)=-ZR(J)
      PRINT 2999
 2999 FORMAT(1H1)
      PRINT 140
140   FORMAT(//20X,32H INVERSE NODE ADMITTANCE MATRIX     /)
      DO 144 KA=1,NODES,4
      IF(KA+3-NODES)142,141,141
141   KB=NODES
      GO TO 143
142   KB=KA+3
143   PRINT 313,(JCH,JCH=KA,KB)
      DO 144 J=1,NODES
144   PRINT 317, J, (ZR(J,K),ZI(J,K), K=KA,KB)
317   FORMAT(I5,4(F17.6,F13.6))
      DO 149 J=1,MD
      DO 149 K=1,MD
147   RR(J,K)=ZR(J,K)
149   RI(J,K)=ZI(J,K)
      PRINT 150
150   FORMAT(43H1IMPEDANCE MATRIX R REGULATED VOLTAGE NODES    //)
      DO 154 KA=1,MD,4
      IF(KA+3-MD)152,151,151
151   KB=MD
      GO TO 153
152   KB=KA+3
153   PRINT 313,(JCH,JCH=KA,KB)
      DO 154 J=1,MD
154   PRINT 317,J,(RR(J,K),RI(J,K),K=KA,KB)
      N=MD
      CALL CMINV(RR,RI,RNR,RNI,MD,IRR)
      PRINT 160
160   FORMAT(//45H ADMITTANCE MATRIX RN REGULATED VOLTAGE NODES    //)
      DO 164 KA=1,MD,4
```

```
      IF(KA+3-MD)162,161,161
161   KB=MD
      GO TO 163
162   KB=KA+3
163   PRINT 313,(JCH,JCH=KA,KB)
      DO 164 J=1,MD
164   PRINT 316,J,(RNR(J,K),RNI(J,K),K=KA,KB)
      BASE=0
      DO 165 J=1,MD
      QD=RNR(J,J)*RNR(J,J)+RNI(J,J)*RNI(J,J)
      BASE=BASE+QD
      ZCR(J)=RNR(J,J)/QD
      ZCI(J)= -RNI(J,J)/QD
165   ZCZ(J)=SQRT(ZCR(J)**2 + ZCI(J)**2)
      BASE=SQRT(MD/BASE)
      PRINT 2999
      PRINT 1
 1    FORMAT(/)
      PRINT 401
401   FORMAT(/37H INTERNAL IMPEDANCE OF POWER NETWORK      /)
      PRINT 404
404   FORMAT(38H              Z=ZCZ       R=ZCR       X=ZCI      )
      PRINT 324, (J,ZCZ(J),ZCR(J),ZCI(J),J=1,MD)
324   FORMAT(I5,3F11.4)
      PRINT 1
      PRINT 418
418   FORMAT(/46H NODE CONNECTION VECTORS OF TRANSMISSION LINES     )
      PRINT 419, (JX,JX=1,L)
419   FORMAT (5H LINE    24I5)
C     READ TABLE DEFINING ORDER OF FAULTED LINES
      READ 327, (JF(LN),LN=1,L)
327   FORMAT (16I5)
      READ 327, (JG(LN),LN=1,L)
      PRINT 328,(JF(LN),LN=1,L)
328   FORMAT (5H NODE 24I5)
      PRINT 328,(JG(LN),LN=1,L)
      RETURN
      END
```

```
      SUBROUTINE CMINV(A,B,C,D,N,IRR)
      DIMENSION A(N,N),B(N,N),C(N,N),D(N,N)
      DIMENSION DUMY(10)
C     DUMY REFERS TO A VECTOR USED WITH VARIABLE DIMENSION IN MTMPB.
      DIMENSION IPIV(10),IND(10,2)
C     IPIV AND IND HAVE VARIABLE DIMENSION IN SUBROUTINE MTINV.
C     A AND B ARE NXN MATRICES CONTAINING THE REAL AND IMAGINARY PARTS
C     C AND D WILL CONTAIN INVERSE.
C     IRR WILL BE NON ZERO IF MATRIX IS SINGULAR
      L=1
    5 DO 10 I=1,N
      DO 10 J=1,N
   10 D(I,J)=-A(I,J)
C     INVERT -A
      CALL MTINV(D,N,C,0,DUM,IRR,IPIV,IND)
C     CHECK A NON -SINGULAR
      IF (IRR)  70,20,70
C     COMPUTE  C=(A+B*A(-1)*B)(-1)
20    CALL MTMPB(B,D,D,N,N,N,DUMY)
      CALL MTMPB(D,B,C,N,N,N,DUMY)
      DO 30 I=1,N
      DO 30 J=1,N
   30 C(I,J)=A(I,J)-C(I,J)
      CALL MTINV(C,N,D,0,DUM,IRR,IPIV,IND)
C     CHECK THAT C EXISTS
      IF(IRR)  40,50,40
   40 IRR=2
      RETURN
C     COMPUTE D=-C*B*A(-1)
50    CALL MTMPB(C,D,D,N,N,N,DUMY)
      GO TO (60,100),L
C     SUCCESSFUL INVERSION
   60 RETURN
C     A IS SINGULAR, INTERCHANGE A AND B AND TRY AGAIN.
   70 DO  80 I=1,N
      DO 80 J=1,N
      DUM=A(I,J)
      A(I,J)=B(I,J)
   80 B(I,J)=DUM
C     IF L=2, A AND B ARE BOTH SINGULAR. IRR=1
      IF(L-2)  90,60,60
   90 L=2
      GO TO 5
C     INTERCHANGE A AND B, C AND D WITH CHANGED SIGNS.
  100 DO  110  I=1,N
      DO  110 J=1,N
      DUM=A(I,J)
      A(I,J)=B(I,J)
      B(I,J)=DUM
      DUM=-C(I,J)
      C(I,J)=-D(I,J)
  110 D(I,J)=DUM
      GO TO 60
      END
```

SUBROUTINE ESPD(K)



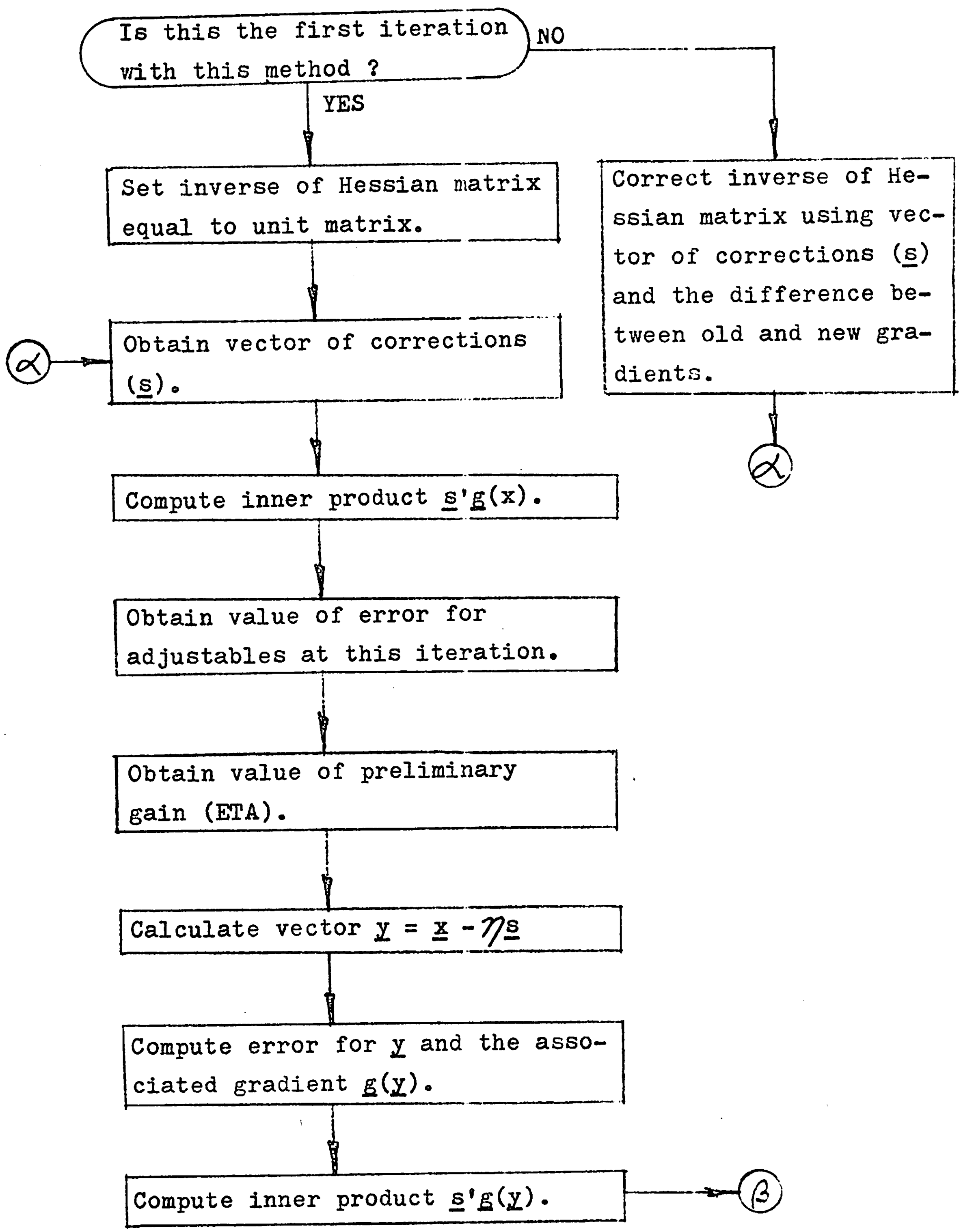

Have all the combinations of pairs of adjustables been taken into account ?
NO
YES
Obtain remaining elements of matrix of second partials using its symetry.
RETURN

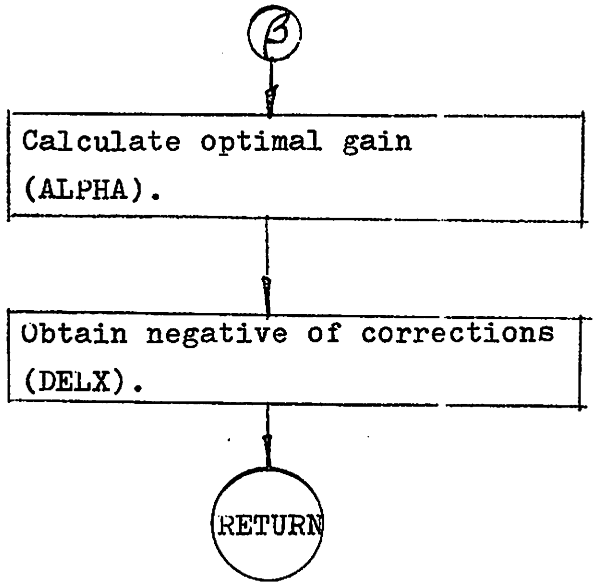
```
      SUBROUTINE ESPD(K)
C     ******************************************************************
C     *THIS SUBROUTINE CALCULATES THE EXACT SECOND PARTIAL DERIVATIVES OF*
C     *THE ERROR FUNCTION, DEFINED IN SUBROUTINE REST, WITH RESPECT TO   *
C     *THE ELEMENTS OF X (ADJUSTABLES). THE OUTPUT IS THE MATRIX OF SEC- *
C     *OND PARTIAL DERIVATIVES, AND IS STORED IN A.                      *
C     ******************************************************************
      COMMON INDEX
      COMMON/SHARE/X(100),DEL(100),A(100,100),N
      COMMON/CRST/DELX(100),DELXO(100)
      COMMON/ADMIT/RNR(8,8),RNI(8,8),MD
      COMMON/COSTS/CP2(8),CQ2(8)
      COMMON/TAR/PTAR(8),QTAR(8),VATAR(8)
      COMMON/DERIV/DPDV(8,8),DPDA(8,8),DQDV(8,8),DQDA(8,8)
      COMMON/ERROR/PE(8),QE(8)
      DIMENSION AP(20,20),AQ(20,20)
      IF(K.LE.2)GO TO 100
      DO 100 M=1,N
      ML=M
      DO 100 L=ML,N
      A(M,L)=0
      DO 200 J=1,MD
      IF(L.LE.MD)2,3
  2   LL=L
      GO TO 4
  3   LL=L-MD
  4   IF(M.LE.MD)5,6
  5   MM=M
      GO TO 7
  6   MM=M-MD
  7   IF(LL.EQ.MM)8,9
  8   IF(J.EQ.LL)10,11
  10  IF(L.LE.MD)12,13
  12  AP(M,L)=2.*RNR(L,L)
      AQ(M,L)=-2.*RNI(L,L)
  20  A(M,L)=A(M,L)+(CP2(J)*(PE(J)*AP(M,L)+DPDV(J,M)*DPDV(J,L))+CQ2(J)*(
     1QE(J)*AQ(M,L)+DQDV(J,M)*DQDV(J,L)))*1./VATAR(J)**2
      GO TO 200
  13  IF(M.LE.MD)14,15
  14  AP(M,L)=0
      AQ(M,L)=0
      DO 16 K=1,MD
      IF(K.EQ.LL)16,17
  17  KA=K+MD
      AP(M,L)=AP(M,L)+X(K)*RNI(LL,K)*COS(X(L)-X(KA))-X(K)*RNR(LL,K)*SIN(
     1X(L)-X(KA))
      AQ(M,L)=AQ(M,L)+RNR(LL,K)*X(K)*COS(X(L)-X(KA))+RNI(LL,K)*X(K)*SIN(
     1X(L)-X(KA))
  16  CONTINUE
  21  A(M,L)=A(M,L)+(CP2(J)*(PE(J)*AP(M,L)+DPDV(J,M)*DPDA(J,LL))+CQ2(J)*
     1(QE(J)*AQ(M,L)+DQDV(J,M)*DGDA(J,LL)))*1./VATAR(J)**2
      GO TO 200
  15  AP(M,L)=0
      AQ(M,L)=0
      DO 18 K=1,MD
```

```
      IF(K.EQ.LL)18,19
 19   KA=K+MD
      AP(M,L)=AP(M,L)+X(LL)*(-RNI(LL,K)*X(K)*SIN(X(L)-X(KA))-RNR(LL,K)*X
     1(K)*COS(X(L)-X(KA)))
      AQ(M,L)=AQ(M,L)+X(LL)*(-RNR(LL,K)*X(K)*SIN(X(L)-X(KA))+RNI(LL,K)*X
     1(K)*COS(X(L)-X(KA)))
 18   CONTINUE
 22   A(M,L)=A(M,L)+(CP2(J)*(PE(J)*AP(M,L)+DPDA(J,MM)*DPDA(J,LL))+CQ2(J)
     1*(QE(J)*AQ(M,L)+DQDA(J,MM)*DQDA(J,LL)))*1./VATAR(J)**2
      GO TO 200
 11   IF(L.LE.MD)23,24
 23   AP(M,L)=0
      AQ(M,L)=0
      GO TO 20
 24   IF(M.LE.MD)25,26
 25   JA=J+MD
      AP(M,L)=X(J)*(RNR(J,LL)*SIN(X(JA)-X(L))-RNI(J,LL)*COS(X(JA)-X(L)))
      AQ(M,L)=-X(J)*(RNR(J,LL)*COS(X(JA)-X(L))+RNI(J,LL)*SIN(X(JA)-X(L))
     1)
      GO TO 21
 26   JA=J+MD
      AP(M,L)=X(J)*X(LL)*(-RNR(J,LL)*COS(X(JA)-X(L))-RNI(J,LL)*SIN(X(JA)
     1-X(L)))
      AQ(M,L)=-X(J)*X(LL)*(RNR(J,LL)*SIN(X(JA)-X(L))-RNI(J,LL)*COS(X(JA)
     1-X(L)))
      GO TO 22
 9    IF(J.EQ.LL)27,28
 27   IF(L.LE.MD)29,30
 29   LA=L+MD
      MA=M+MD
      AP(M,L)=RNR(L,M)*COS(X(LA)-X(MA))+RNI(L,M)*SIN(X(LA)-X(MA))
      AQ(M,L)=RNR(L,M)*SIN(X(LA)-X(MA))-RNI(L,M)*COS(X(LA)-X(MA))
      GO TO 20
 30   IF(M.LE.MD)31,32
 31   MA=M+MD
      AP(M,L)=X(LL)*(RNI(LL,M)*COS(X(L)-X(MA))-RNR(LL,M)*SIN(X(L)-X(MA))
     1)
      AQ(M,L)=X(LL)*(RNR(LL,M)*COS(X(L)-X(MA))+RNI(LL,M)*SIN(X(L)-X(MA))
     1)
      GO TO 21
 32   AP(M,L)=X(LL)*X(MM)*(RNI(LL,MM)*SIN(X(L)-X(M))+RNR(LL,MM)*COS(X(L)
     1-X(M)))
      AQ(M,L)=X(LL)*X(MM)*(RNR(LL,MM)*SIN(X(L)-X(M))-RNI(LL,MM)*COS(X(L)
     1-X(M)))
      GO TO 22
 28   IF(J.EQ.MM)33,34
 33   IF(L.LE.MD)35,36
 35   JA=J+MD
      LA=L+MD
      AP(M,L)=RNR(J,L)*COS(X(JA)-X(LA))+RNI(J,L)*SIN(X(JA)-X(LA))
      AQ(M,L)=RNR(J,L)*SIN(X(JA)-X(LA))-RNI(J,L)*COS(X(JA)-X(LA))
      GO TO 20
 36   IF(M.LE.MD)37,38
 37   JA=J+MD
      AP(M,L)=X(LL)*(RNR(J,LL)*SIN(X(JA)-X(L))-RNI(J,LL)*COS(X(JA)-X(L))
```

```
     1)
      AQ(M,L)=-X(LL)*(RNR(J,LL)*COS(X(JA)-X(L))+RNI(J,LL)*SIN(X(JA)-X(L)
     1))
      GO TO 21
 38   JA=J+MD
      AP(M,L)=X(J)*X(LL)*(RNR(J,LL)*COS(X(JA)-X(L))+RNI(J,LL)*SIN(X(JA)-
     1X(L)))
      AQ(M,L)=-X(J)*X(LL)*(-RNR(J,LL)*SIN(X(JA)-X(L))+RNI(J,LL)*COS(X(JA
     1)-X(L)))
      GO TO 22
 34   IF(L.LE.MD)39,40
 39   AP(M,L)=0
      AQ(M,L)=0
      GO TO 20
 40   IF(M.LE.MD)41,42
 41   AP(M,L)=0
      AQ(M,L)=0
      GO TO 21
 42   AP(M,L)=0
      AQ(M,L)=0
      GO TO 22
 200  CONTINUE
      A(M,L)=2.*A(M,L)
      A(L,M)=A(M,L)
 100  CONTINUE
      RETURN
      END
```

SUBROUTINE FPOWELL(ICOUNT,TOLER,ALPHA)

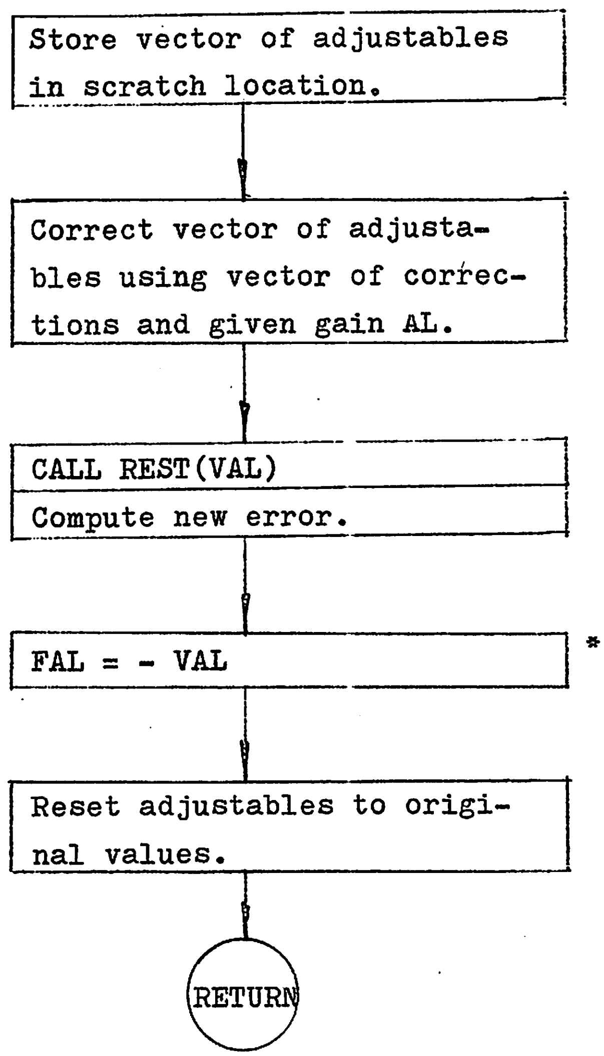



β
Calculate optimal gain (ALPHA).
Obtain negative of corrections (DELX).
RETURN

```
      SUBROUTINE FPOWELL(ICOUNT,TOLER,ALPHA)
C     *******************************************************************
C     *THIS SUBROUTINE COMPUTES THE CORRECTIONS FOR VECTOR X BY MEANS OF *
C     *FLETCHER AND POWELLS METHOD.  THE NEGATIVE OF THE CORRECTIONS     *
C     *IS STORED IN DELX AND THE OPTIMAL GAIN IS OBTAINED BY A CUBICAL   *
C     *INTERPOLATION AND STORED IN ALPHA.                                *
C     *******************************************************************
      COMMON/SHARE/X(100),DEL(100),A(100,100),N
      COMMON/CRST/DELX(100),DELXO(100)
      COMMON/FLET/G(20),S(20),AC(20,20),BC(20,20),ICOUNTF
      ICOUNTF=ICOUNTF+1
      IF(ICOUNTF.EQ.1)1,2
1     DO 3 I=1,N
      DO 3 J=1,N
      IF(I.EQ.J)4,5
4     A(I,J)=1.
      GO TO 3
5     A(I,J)=0.
3     CONTINUE
      GO TO 24
2     DO 16 I=1,N
16    G(I)=DELX(I)-G(I)
      DO 17 I=1,N
      DELXO(I)=0.
      DO 17 J=1,N
17    DELXO(I)=DELXO(I)+A(I,J)*G(J)
      PRODB=0.
      DO 18 I=1,N
18    PRODB=PRODB+G(I)*DELXO(I)
      PRODA=0.
      DO 19 I=1,N
19    PRODA=PRODA+DEL(I)*G(I)
      DO 20 I=1,N
      DO 20 J=1,N
20    AC(I,J)=DEL(I)*DEL(J)/PRODA
      DO 21 I=1,N
      DEL(I)=0.
      DO 21 J=1,N
21    DEL(I)=DEL(I)+A(J,I)*G(J)
      DO 22 I=1,N
      DO 22 J=1,N
22    BC(I,J)=-DELXO(I)*DEL(J)/PRODB
      DO 23 I=1,N
      DO 23 J=1,N
23    A(I,J)=A(I,J)+AC(I,J)+BC(I,J)
24    DO 6 I=1,N
      S(I)=0.
      DO 6 J=1,N
6     S(I)=S(I)+A(I,J)*DELX(J)
      DO 7 I=1,N
7     S(I)=-S(I)
      PRODX=0.
      DO 8 I=1,N
8     PRODX=PRODX+S(I)*DELX(I)
      CALL REST(EX)
```

```
      ETA=-2.*EX/PRODX
      ETA=AMIN1(1.,ETA)
      DO 9 I=1,N
 9    DEL(I)=X(I)
      DO 10 I=1,N
 10   X(I)=X(I)+ETA*S(I)
      CALL REST(EY)
      DO 13 I=1,N
 13   G(I)=DELX(I)
      CALL GRAD1
      DO 12 I=1,N
 12   X(I)=DEL(I)
      PRODY=0.
      DO 11 I=1,N
 11   PRODY=PRODY+DELX(I)*S(I)
      DO 15 I=1,N
 15   DELX(I)=-S(I)
      Z=(3./ETA)*(EX-EY)+PRODX+PRODY
      DISC=Z**2-PRODX*PRODY
      IF(DISC.LT.0.)GO TO 30
      W=SQRT(DISC)
      ALPHA=ETA*(1.-(PRODY+W-Z)/(PRODY-PRODX+2.*W))
      GO TO 31
 30   CALL INTER(AA,BB)
      PRINT 32
 32   FORMAT(/* ODS IS PERFORMED SINCE DISCRIMINANT IS NEGATIVE*/)
      CALL ODS(AA,BB,TOLER,100,ALPHA,FMAX)
 31   DO 14 I=1,N
 14   DEL(I)=ALPHA*S(I)
      RETURN
      END
```

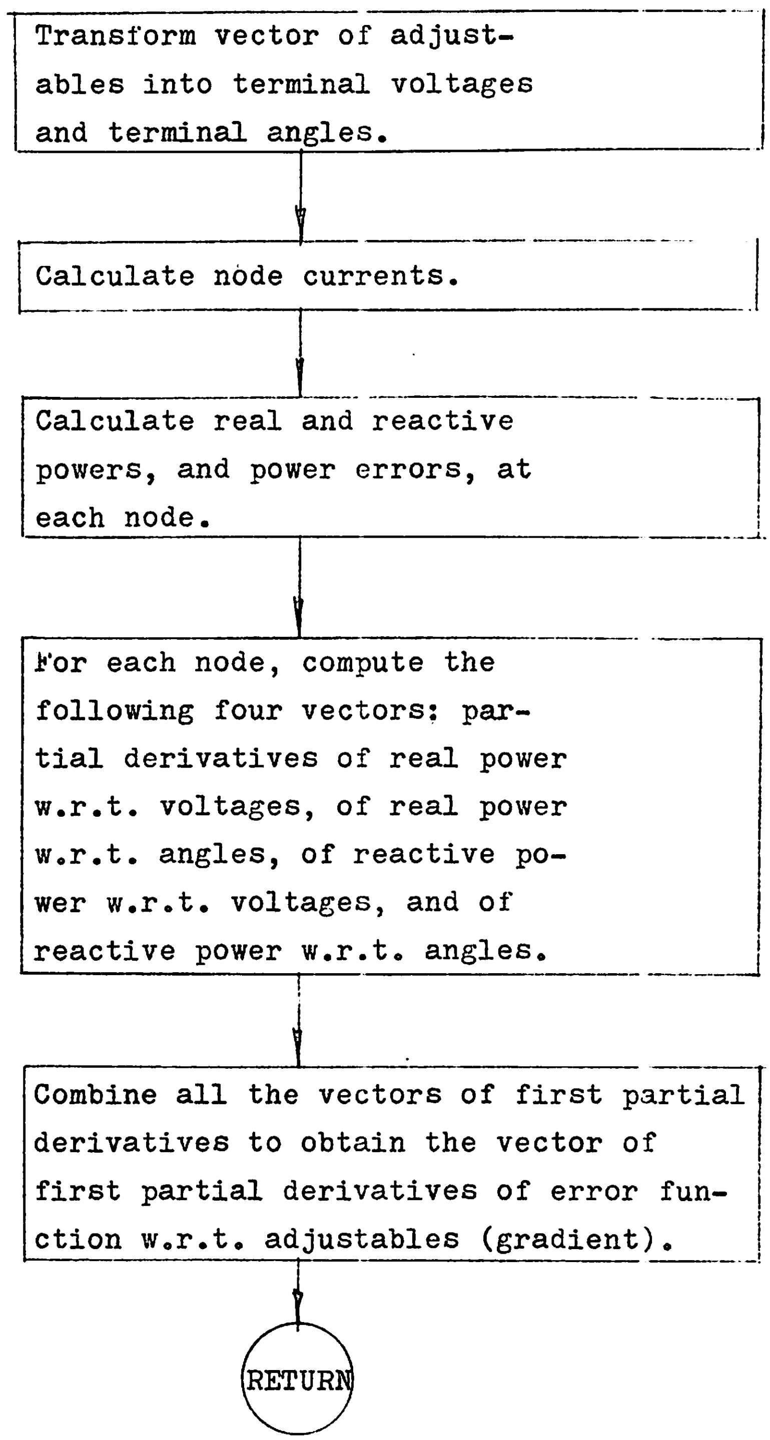
SUBROUTINE FUNCT(AL,FAL)



* The minus sign appears because subroutines INTER and ODS are written for maximizing and here we want to minimize.

```
      SUBROUTINE FUNCT(AL,FAL)
C     ******************************************************************
C     *THIS IS AN AUXILIARY SUBROUTINE FOR THE ONE-DIMENSIONAL SEARCH.  *
C     *THIS SUBROUTINE CALCULATES THE ERROR FOR DIFFERENTS VALUES OF THE *
C     *GAIN LAMBDA.                                                      *
C     ******************************************************************
      COMMON INDEX
      COMMON/SHARE/X(100),DEL(100),A(100,100),N
      COMMON/CRST/DELX(100),DELX0(100)
      DO 20 J=1,N
      DEL(J)=X(J)
   20 X(J)=X(J)-AL*DELX(J)
      CALL REST(VAL)
      FAL=-1.*VAL
      DO 30 J=1,N
   30 X(J)=DEL(J)
      RETURN
      END
```

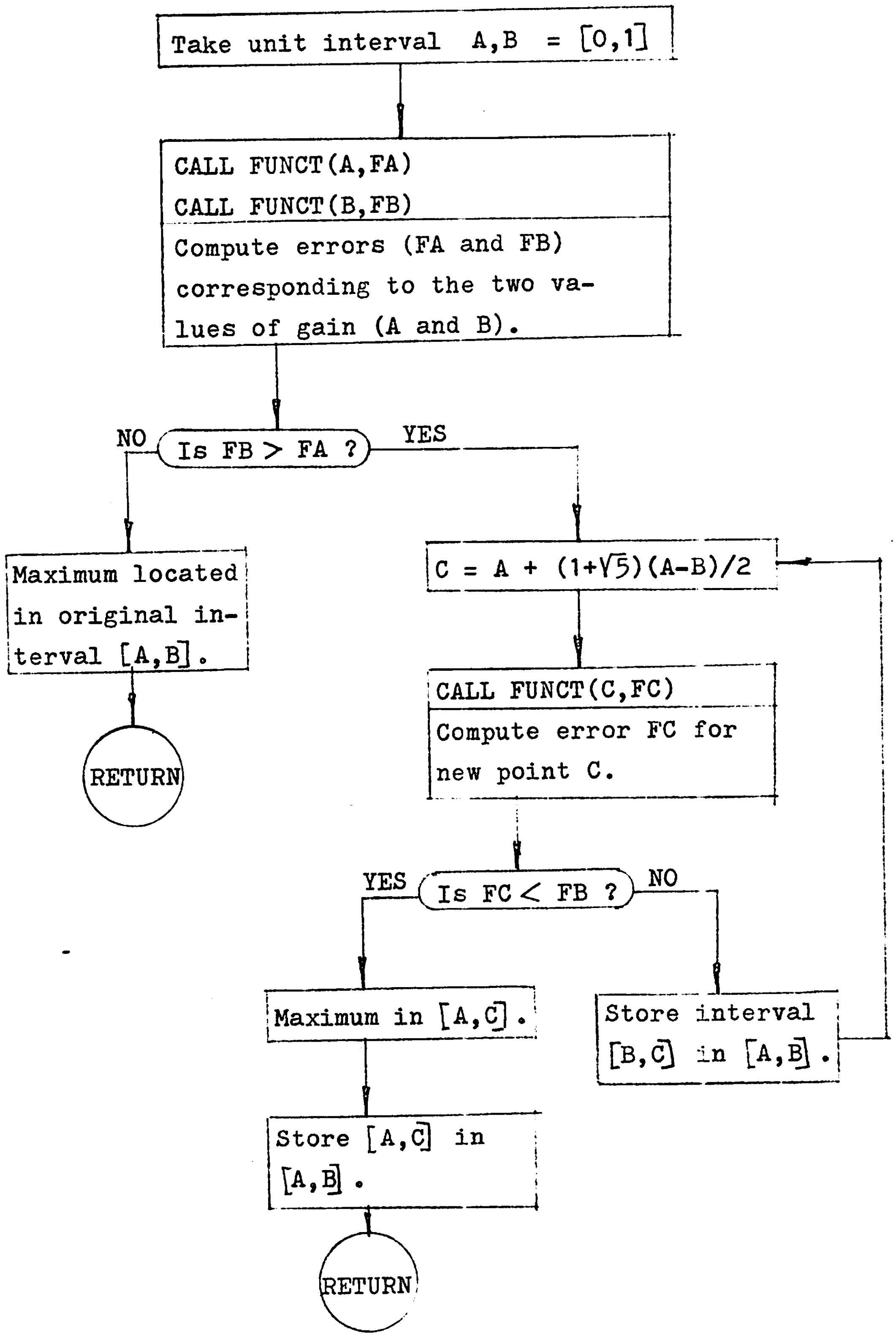
SUBROUTINE GRAD1



```
      SUBROUTINE GRAD1
C     ***********************************************************************
C     *THIS SUBROUTINE CALCULATES THE VECTOR OF FIRST PARTIAL DERIVATI-      *
C     *VES OF ERROR FUNCTION WITH RESPECT TO THE VECTOR X. THE OUTPUT        *
C     *(GRADIENT) IS STORED IN DELX AND DELXO. ALSO FIRST PARTIALS OF P      *
C     *AND Q W.R.T. VOLTAGES AND ANGLES ARE STORED IN THE COMMON BLOCK       *
C     *DERIV, AND POWER ERRORS, PE AND QE, IN COMMON BLOCK ERROR.            *
C     ***********************************************************************
      COMMON/SHARE/X(100),DEL(100),A(100,100),N
      COMMON/CRST/DELX(100),DELX0(100)
      COMMON INDEX
      COMMON/ADMIT/RNR(8,8),RNI(8,8),MD
      COMMON/COSTS/CP2(8),CQ2(8)
      COMMON/TAR/PTAR(8),QTAR(8),VATAR(8)
      COMMON/DERIV/DPDV(8,8),DPDA(8,8),DQDV(8,8),DQDA(8,8)
      COMMON/ERROR/PE(8),QE(8)
      DIMENSION VREF(8),ANGTER(8),VR(8),VI(8),UMR(8),UMI(8)
      DIMENSION ANGDIFF(8,8)
      DO 20 J=1,MD
      VREF(J)=X(J)
      JA=J+MD
   20 ANGTER(J)=X(JA)
      DO 206 J=1,MD
      VR(J)=VREF(J)*COS(ANGTER(J))
  206 VI(J)=VREF(J)*SIN(ANGTER(J))
      DO 207 J=1,MD
      UMR(J)=0
      UMI(J)=0
      DO 207 K=1,MD
      UMR(J)=UMR(J)+RNR(J,K)*VR(K)-RNI(J,K)*VI(K)
  207 UMI(J)=UMI(J)+RNR(J,K)*VI(K)+RNI(J,K)*VR(K)
      DO 511 J=1,MD
      VAR=VI(J)*UMR(J)-VR(J)*UMI(J)
      QE(J)=VAR-QTAR(J)
      POWER=VR(J)*UMR(J)+VI(J)*UMI(J)
  511 PE(J)=POWER-PTAR(J)
      DO 1210 J=1,MD
      DO 1210 K=1,MD
 1210 ANGDIFF(J,K)=ANGTER(J)-ANGTER(K)
      DO 1230  J=1,MD
      DO 1230  K=1,MD
      IF(J.EQ.K)1230,1215
 1215 DPDV(J,K)=VREF(J)*(RNR(J,K)*COS(ANGDIFF(J,K))+RNI(J,K)*SIN(ANGDIFF(J
     1(J,K)))
      DQDV(J,K)=VREF(J)*(RNR(J,K)*SIN(ANGDIFF(J,K))-RNI(J,K)*COS(ANGDIFF(J
     1(J,K)))
      DPDA(J,K)=VREF(K)*DQDV(J,K)
      DQDA(J,K)=-VREF(K)*DPDV(J,K)
 1230 CONTINUE
C     DIAGONAL TERMS ARE CALCULATED NEXT.
      DO 1245  J=1,MD
      K=J
      DPDV(J,J)=VREF(J)*RNR(J,J)
      DQDV(J,J)=-VREF(J)*RNI(J,J)
      DO 1240 L=1,MD
```

```
      DPDV(J,J)=DPDV(J,J)+VREF(L)*(RNR(J,L)*COS(ANGDIFF(J,L))+RNI(J,L)*S
     1IN(ANGDIFF(J,L)))
      DQDV(J,J)=DQDV(J,J)+VREF(L)*(RNR(J,L)*SIN(ANGDIFF(J,L))-RNI(J,L)*COS(
     1OS(ANGDIFF(J,L)))
 1240 CONTINUE
      DPDA(J,J)=-VREF(J)*(DQDV(J,J)+2.0*VREF(J)*RNI(J,J))
      DQDA(J,J)=VREF(J)*(DPDV(J,J)-2.0*VREF(J)*RNR(J,J))
 1245 CONTINUE
      DO 30 L=1,N
      DELX(L)=0
      DO 30 J=1,MD
      IF(L.LE.MD)31,32
   31 DELX(L)=(2.*CP2(J)*PE(J)*DPDV(J,L)+2.*CQ2(J)*QE(J)*DQDV(J,L))/VATA
     1R(J)**2+DELX(L)
      GO TO 30
   32 LA=L-MD
      DELX(L)=(2.*CP2(J)*PE(J)*DPDA(J,LA)+2.*CQ2(J)*QE(J)*DQDA(J,LA))/VA
     1TAR(J)**2+DELX(L)
   30 CONTINUE
      DO 400 J=1,N
  400 DELXO(J)=DELX(J)
      RETURN
      END
```

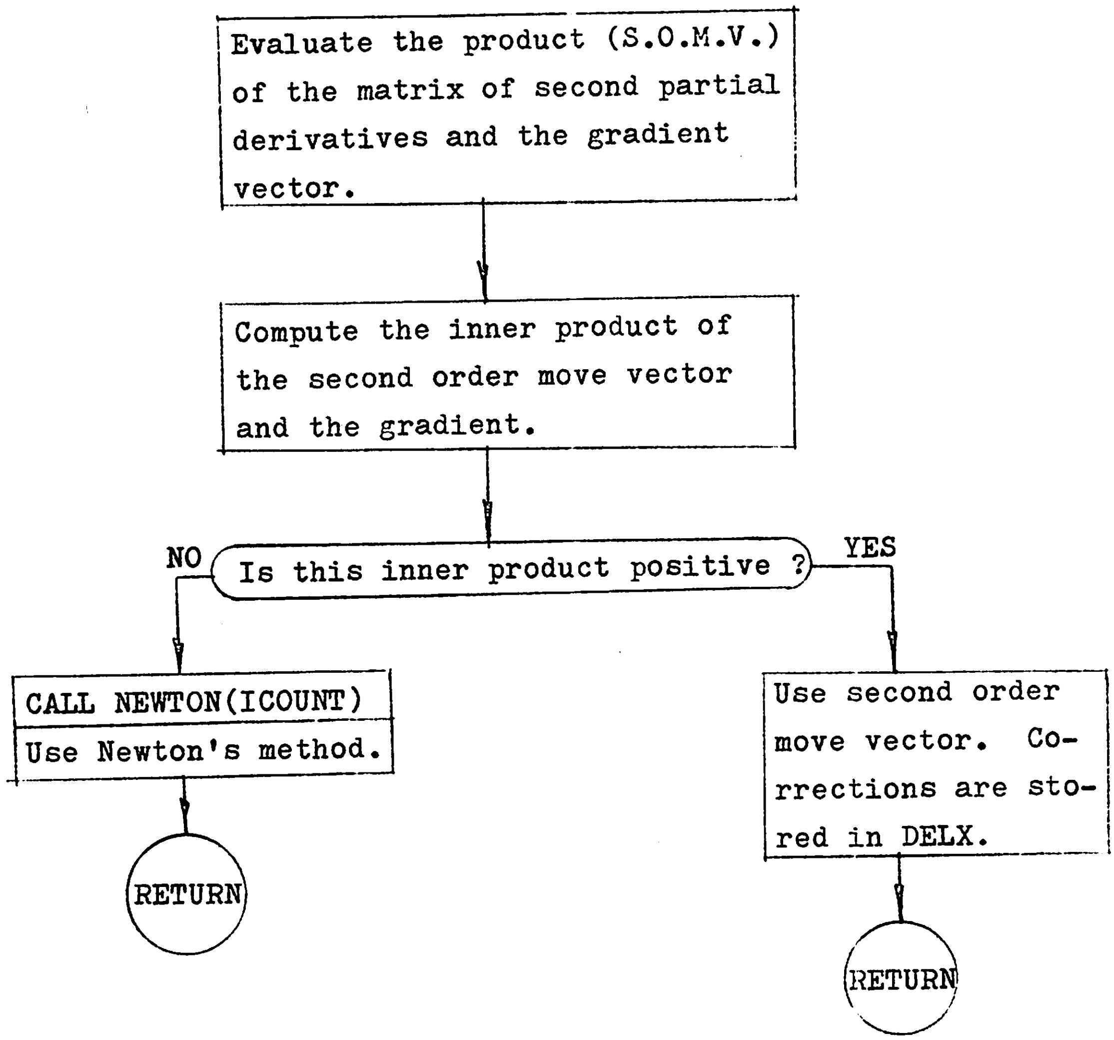
SUBROUTINE INTER(A,B)*



* This subroutine is actually written to accomplish maximization. To minimize see footnote in subroutine FUNCT.

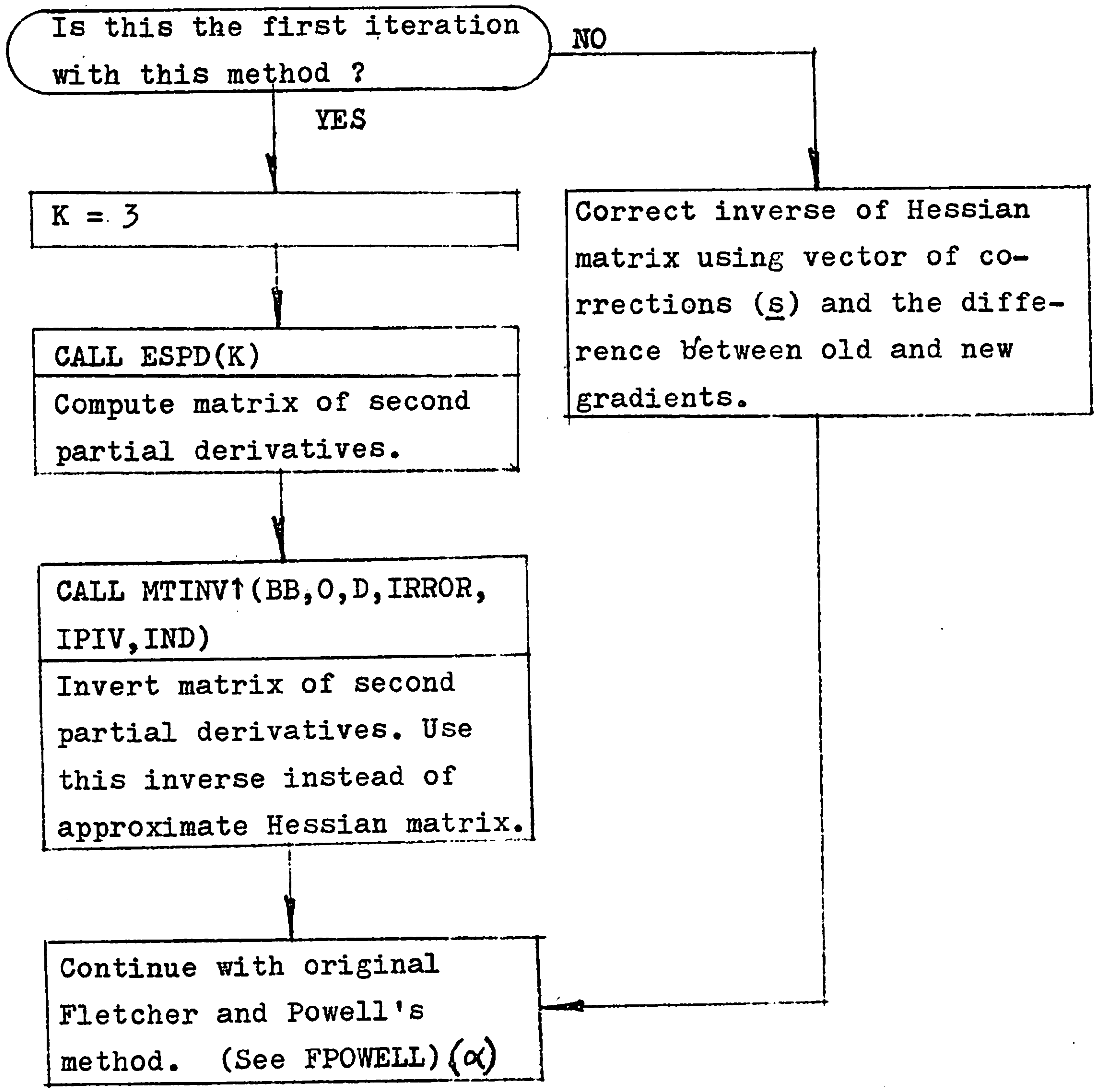
```
      SUBROUTINE INTER(A,B)
C     ************************************************************************
C     *THIS IS AN AUXILIARY SUBROUTINE FOR ONE-DIMENSIONAL SEARCH. THE    *
C     *INITIAL INTERVAL FOR THE SEARCH IS FOUND.                           *
C     ************************************************************************
      T=0.5+SQRT(5.)/2.
      A=0.
      B=1.
      CALL FUNCT(A,FA)
      CALL FUNCT(B,FB)
      IF(FB.GT.FA) GO TO 20
      RETURN
   20 C=A+T*(B-A)
      CALL FUNCT(C,FC)
      IF(FC.GT.FB)  GO TO 30
      B=C
      RETURN
   30 A=B
      FA=FB
      B=C
      FB=FC
      GO TO 20
      END
```

SUBROUTINE METHOD(IRROR,ICOUNT)



```
      SUBROUTINE METHOD(IRROR,ICOUNT)
C     ******************************************************************
C     *THIS SUBROUTINE CHECKS IF IT IS POSSIBLE TO USE THE SECOND ORDER *
C     *MOVE VECTOR, THAT IS, IF MATRIX OF SECOND PARTIAL DERIVATIVES IS *
C     *POSITIVE DEFINITE. IF NOT POSSIBLE, NEWTONS METHOD IS USED.      *
C     ******************************************************************
      COMMON INDEX
      COMMON/SHARE/X(100),DEL(100),A(100,100),N
      COMMON/CRST/DELX(100),DELX0(100)
      IF(IRROR.EQ.1)GO TO 80
      DO 30 I=1,N
      DELX(I)=0.
      DO 30 J=1,N
30    DELX(I)=DELX(I)+A(I,J)*DELX0(J)
      PROD=0.
      DO 40 I=1,N
40    PROD=PROD+DELX(I)*DELX0(I)
      IF(PROD.GT.0.)93,80
80    CALL NEWTON(ICOUNT)
93    CONTINUE
      RETURN
      END
```

SUBROUTINE MODFPOW(ICOUNT,TOLER,ALPHA)



```
      SUBROUTINE MODFPOW(ICOUNT,TOLER,ALPHA)
C     ******************************************************************
C     *THIS SUBROUTINE COMPUTES THE CORRECTIONS FOR VECTOR X BY MEANS OF *
C     *A MODIFICATION TO FLETCHER AND POWELLS METHOD.  THIS MODIFICA-    *
C     *TION CONSISTS OF USING THE MATRIX OF EXACT SECOND PARTIAL DERIVA- *
C     *TIVES AS THE HESSIAN MATRIX FOR THE FIRST ITERATION, INSTEAD OF   *
C     *THE UNIT MATRIX. THE NEGATIVE OF THE CORRECTIONS IS STORED IN DELX*
C     *AND THE OPTIMAL GAIN IS OBTAINED BY A CUBICAL INTERPOLATION AND   *
C     *STORED IN ALPHA.                                                  *
C     ******************************************************************
      COMMON/SHARE/X(100),DEL(100),A(100,100),N
      COMMON/CRST/DELX(100),DELX0(100)
      COMMON/FLET/G(20),S(20),AC(20,20),BC(20,20),ICOUNTF
      DIMENSION BB(16,1),IPIV(16),IND(16,2)
      ICOUNTF=ICOUNTF+1
      IF(ICOUNTF.EQ.1)1,2
 1    K=3
      PRINT 2002
 2002 FORMAT(* MODIFIED FLETCHER AND POWELLS METHOD IS USED NEXT*/)
      CALL ESPD(K)
      DO 8001 I=1,N
 8001 BB(I,1)=1.0
      CALL MTINV1(BB,0,D,IRROR,IPIV,IND)
      GO TO 24
 2    DO 16 I=1,N
 16   G(I)=DELX(I)-G(I)
      DO 17 I=1,N
      DELX0(I)=0.
      DO 17 J=1,N
 17   DELX0(I)=DELX0(I)+A(I,J)*G(J)
      PRODB=0.
      DO 18 I=1,N
 18   PRODB=PRODB+G(I)*DELX0(I)
      PRODA=0.
      DO 19 I=1,N
 19   PRODA=PRODA+DEL(I)*G(I)
      DO 20 I=1,N
      DO 20 J=1,N
 20   AC(I,J)=DEL(I)*DEL(J)/PRODA
      DO 21 I=1,N
      DEL(I)=0.
      DO 21 J=1,N
 21   DEL(I)=DEL(I)+A(J,I)*G(J)
      DO 22 I=1,N
      DO 22 J=1,N
 22   BC(I,J)=-DELX0(I)*DEL(J)/PRODB
      DO 23 I=1,N
      DO 23 J=1,N
 23   A(I,J)=A(I,J)+AC(I,J)+BC(I,J)
 24   DO 6 I=1,N
      S(I)=0.
      DO 6 J=1,N
 6    S(I)=S(I)+A(I,J)*DELX(J)
      DO 7 I=1,N
 7    S(I)=-S(I)
```

```
      PRODX=0.
      DO 8 I=1,N
    8 PRODX=PRODX+S(I)*DELX(I)
      CALL REST(EX)
      ETA=-2.*EX/PRODX
      ETA=AMIN1(1.,ETA)
      DO 9 I=1,N
    9 DEL(I)=X(I)
      DO 10 I=1,N
   10 X(I)=X(I)+ETA*S(I)
      CALL REST(EY)
      DO 13 I=1,N
   13 G(I)=DELX(I)
      CALL GRAD1
      DO 12 I=1,N
   12 X(I)=DEL(I)
      PRODY=0.
      DO 11 I=1,N
   11 PRODY=PRODY+DELX(I)*S(I)
      DO 15 I=1,N
   15 DELX(I)=-S(I)
      Z=(3./ETA)*(EX-EY)+PRODX+PRODY
      DISC=Z**2-PRODX*PRODY
      IF(DISC.LT.0.)GO TO 30
      W=SQRT(DISC)
      ALPHA=ETA*(1.-(PRODY+W-Z)/(PRODY-PRODX+2.*W))
      GO TO 31
   30 CALL INTER(AA,BB)
      PRINT 32
   32 FORMAT(/* ODS IS PERFORMED SINCE DISCRIMINANT IS NEGATIVE*/)
      CALL ODS(AA,BB,TOLER,100,ALPHA,FMAX)
   31 DO 14 I=1,N
   14 DEL(I)=ALPHA*S(I)
      RETURN
      END
```

```
      SUBROUTINE MTINV(A,N,B,L,D,IRROR,IPIV,IND)
      DIMENSION A(N,N),B(N,1),IPIV(N),IND(N,2)
C     A IS AN NXN MATRIX TO BE INVERTED,OR CONTAINING EQUATION COEFFS
C     B IS AN NXM RHS MATRIX FOR EQUATIONS
C     IF L=0,INVERSE ONLY GIVEN.L POSITIVE,SOLUTIONS ONLY.L NEGATIVE
C      BOTH.   M=ABS(L).
C     D CONTAINS THE DETERMINANT OF THE A MATRIX ON EXIT
C     A IS REPLACED BY THE INVERSE ,B BY THE SOLUTIONS.
C     METHOD OF GAUSS-JORDON PIVOTAL ELIMINATION
      M=IABS(L)
      D=1.0
      DO 10 I=1,N
   10 IPIV(I)=0
      DO 220 I=1,N
      AMAX=0.0
C     SEARCH SUB-MATRIX FOR LARGEST ELEMENT AS PIVOT
      DO  70 J=1,N
      IF(IPIV(J)) 80,30,70
   30 DO  60 K=1,N
      IF(IPIV(K)-1) 40,60,80
C     THIS ROW COLUMN HAS NOT BEEN A PIVOT
   40 IF(ABS (A(J,K))-AMAX) 60,60,50
   50 IROW=J
      ICOL=K
      AMAX=ABS (A(J,K))
   60 CONTINUE
   70 CONTINUE
C     PIVOT FOUND
      IPIV(ICOL)=IPIV(ICOL)+1
      IF(AMAX-1.0E-90 ) 80,80,90
C     MATRIX SINGULAR,ERROR RETURN
   80 IRROR=1
      RETURN
   90 IF(IROW-ICOL) 95,130,95
C     MAKE PIVOT A DIAGONAL ELEMENT BY ROW INTERCHANGE.
   95 D=-D
      DO 100 K=1,N
      AMAX=A(IROW,K)
      A(IROW,K)=A(ICOL,K)
  100 A(ICOL,K)=AMAX
      IF(M) 130,130,110
  110 DO 120 K=1,M
      AMAX=B(IROW,K)
      B(IROW,K)=B(ICOL,K)
  120 B(ICOL,K)=AMAX
  130 IND(I,1)=IROW
      IND(I,2)=ICOL
      AMAX=A(ICOL,ICOL)
      D=D*AMAX
      A(ICOL,ICOL)=1.0
C     DIVIDE PIVOT ROW BY PIVOT
      DO 140 K=1,N
  140 A(ICOL,K)=A(ICOL,K)/AMAX
      IF(M) 170,170,150
  150 DO 160 K=1,M
```

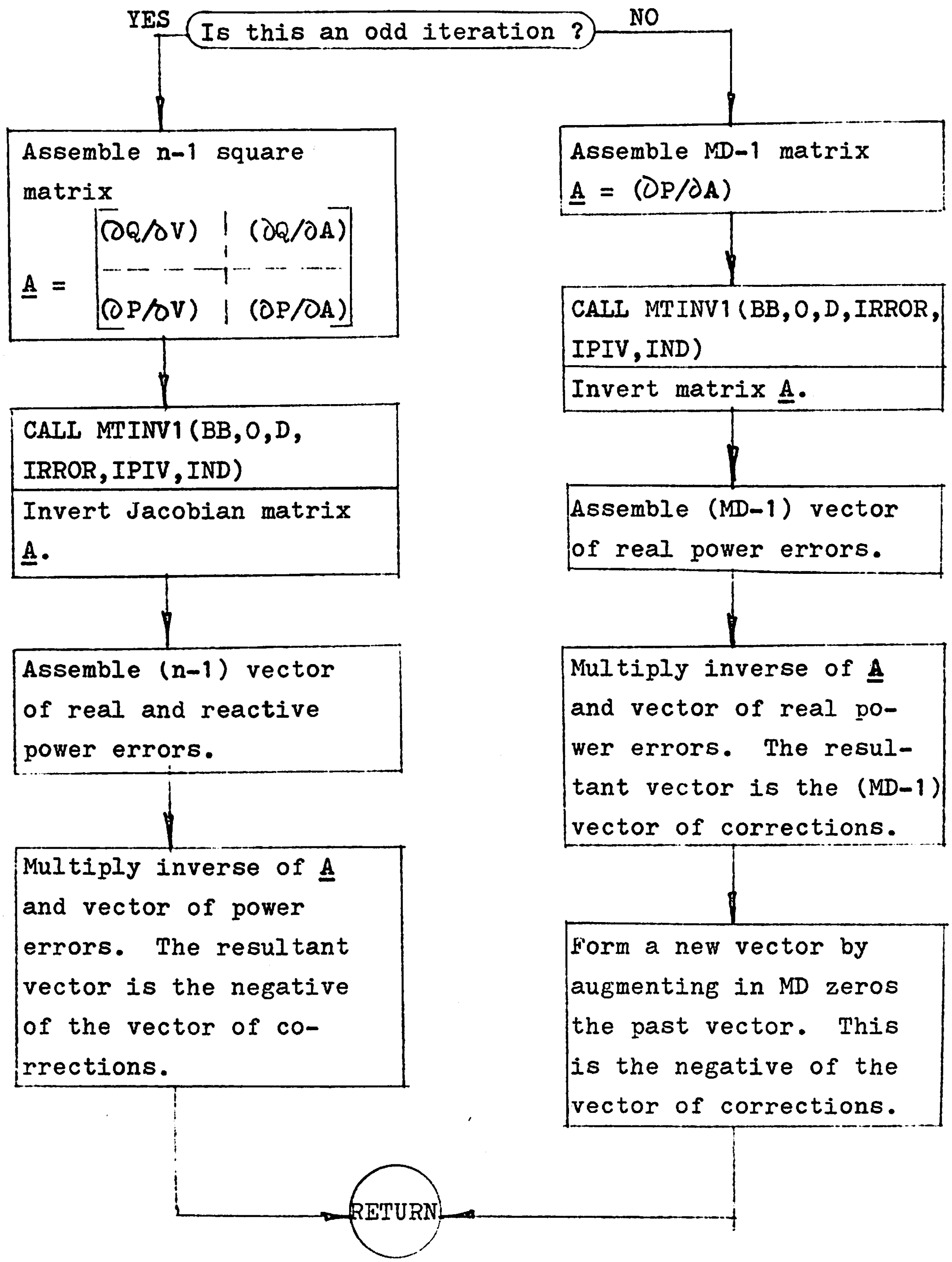
```
  160 B(ICOL,K)=B(ICOL,K)/AMAX
C     REDUCE NON-PIVOT ROWS
  170 DO 220 J=1,N
      IF(J-ICOL) 180,220,180
  180 AMAX=A(J,ICOL)
      A(J,ICOL)=0.0
      DO 190  K=1,N
  190 A(J,K)=A(J,K)-A(ICOL,K)*AMAX
      IF(M)  220,220,200
  200 DO  210 K=1,M
  210 B(J,K)=B(J,K)-B(ICOL,K)*AMAX
  220 CONTINUE
C     AFTER N PIVOTAL CONDENSATIONS,SOLUTIONS LIE IN B MATRIX
      IF(L) 230,230,270
C     FOR INVERSE OF A, INTERCHANGE COLUMNS
  230 DO 260 I=1,N
      J=N+1-I
      IF(IND(J,1)-IND(J,2)) 240,260,240
  240 IROW=IND(J,1)
      ICOL=IND(J,2)
      DO 250  K=1,N
      AMAX=A(K,IROW)
      A(K,IROW)=A(K,ICOL)
  250 A(K,ICOL)=AMAX
  260 CONTINUE
  270 IRROR=0
      RETURN
      END
```

```
      SUBROUTINE MTINV1(B,L,D,IRROR,IPIV,IND)
      COMMON/SHARE/X(100),DEL(100),A(100,100),N
      DIMENSION B(16,1),IPIV(16),IND(16,2)
C     A IS AN NXN MATRIX TO BE INVERTED,OR CONTAINING EQUATION COEFFS
C     B IS AN NXM RHS MATRIX FOR EQUATIONS
C     IF L=0,INVERSE ONLY GIVEN.L POSITIVE,SOLUTIONS ONLY.L NEGATIVE
C      BOTH.   M=ABS(L).
C     D CONTAINS THE DETERMINANT OF THE A MATRIX ON EXIT
C     A IS REPLACED BY THE INVERSE ,B BY THE SOLUTIONS.
C     METHOD OF GAUSS-JORDON PIVOTAL ELIMINATION
      M=IABS(L)
      D=1.0
      DO 10 I=1,N
   10 IPIV(I)=0
      DO 220 I=1,N
      AMAX=0.0
C     SEARCH SUB-MATRIX FOR LARGEST ELEMENT AS PIVOT
      DO  70 J=1,N
      IF(IPIV(J)) 80,30,70
   30 DO  60 K=1,N
      IF(IPIV(K)-1) 40,60,80
C     THIS ROW COLUMN HAS NOT BEEN A PIVOT
   40 IF(ABS (A(J,K))-AMAX) 60,60,50
   50 IROW=J
      ICOL=K
      AMAX=ABS (A(J,K))
   60 CONTINUE
   70 CONTINUE
C     PIVOT FOUND
      IPIV(ICOL)=IPIV(ICOL)+1
      IF(AMAX-1.0E-90 ) 80,80,90
C     MATRIX SINGULAR,ERROR RETURN
   80 IRROR=1
      RETURN
   90 IF(IROW-ICOL) 95,130,95
C     MAKE PIVOT A DIAGONAL ELEMENT BY ROW INTERCHANGE.
   95 D=-D
      DO 100 K=1,N
      AMAX=A(IROW,K)
      A(IROW,K)=A(ICOL,K)
  100 A(ICOL,K)=AMAX
      IF(M) 130,130,110
  110 DO 120 K=1,M
      AMAX=B(IROW,K)
      B(IROW,K)=B(ICOL,K)
  120 B(ICOL,K)=AMAX
  130 IND(I,1)=IROW
      IND(I,2)=ICOL
      AMAX=A(ICOL,ICOL)
      D=D*AMAX
      A(ICOL,ICOL)=1.0
C     DIVIDE PIVOT ROW BY PIVOT
      DO 140 K=1,N
  140 A(ICOL,K)=A(ICOL,K)/AMAX
      IF(M) 170,170,150
```

```
  150 DO 160 K=1,M
  160 B(ICOL,K)=B(ICOL,K)/AMAX
C     REDUCE NON-PIVOT ROWS
  170 DO 220 J=1,N
      IF(J-ICOL) 180,220,180
  180 AMAX=A(J,ICOL)
      A(J,ICOL)=0.0
      DO 190  K=1,N
  190 A(J,K)=A(J,K)-A(ICOL,K)*AMAX
      IF(M)  220,220,200
  200 DO  210 K=1,M
  210 B(J,K)=B(J,K)-B(ICOL,K)*AMAX
  220 CONTINUE
C     AFTER N PIVOTAL CONDENSATIONS,SOLUTIONS LIE IN B MATRIX
      IF(L) 230,230,270
C     FOR INVERSE OF A, INTERCHANGE COLUMNS
  230 DO 260 I=1,N
      J=N+1-I
      IF(IND(J,1)-IND(J,2)) 240,260,240
  240 IROW=IND(J,1)
      ICOL=IND(J,2)
      DO 250  K=1,N
      AMAX=A(K,IROW)
      A(K,IROW)=A(K,ICOL)
  250 A(K,ICOL)=AMAX
  260 CONTINUE
  270 IRROR=0
      RETURN
      END
```

```
      SUBROUTINE MTMPB(A,B,C,M,N,L,D)
      DIMENSION A(N,N),B(N,N),C(N,N),D(N)
C     B AND C MAY BE THE SAME ARRAY.   FORMS  C=A*B
      DO220  I=1,L
      DO210  J=1,M
      D(J)=0.0
      DO210  K=1,N
  210 D(J)=D(J)+A(J,K)*B(K,I)
      DO220  J=1,M
  220 C(J,I)=D(J)
      RETURN
      END
```

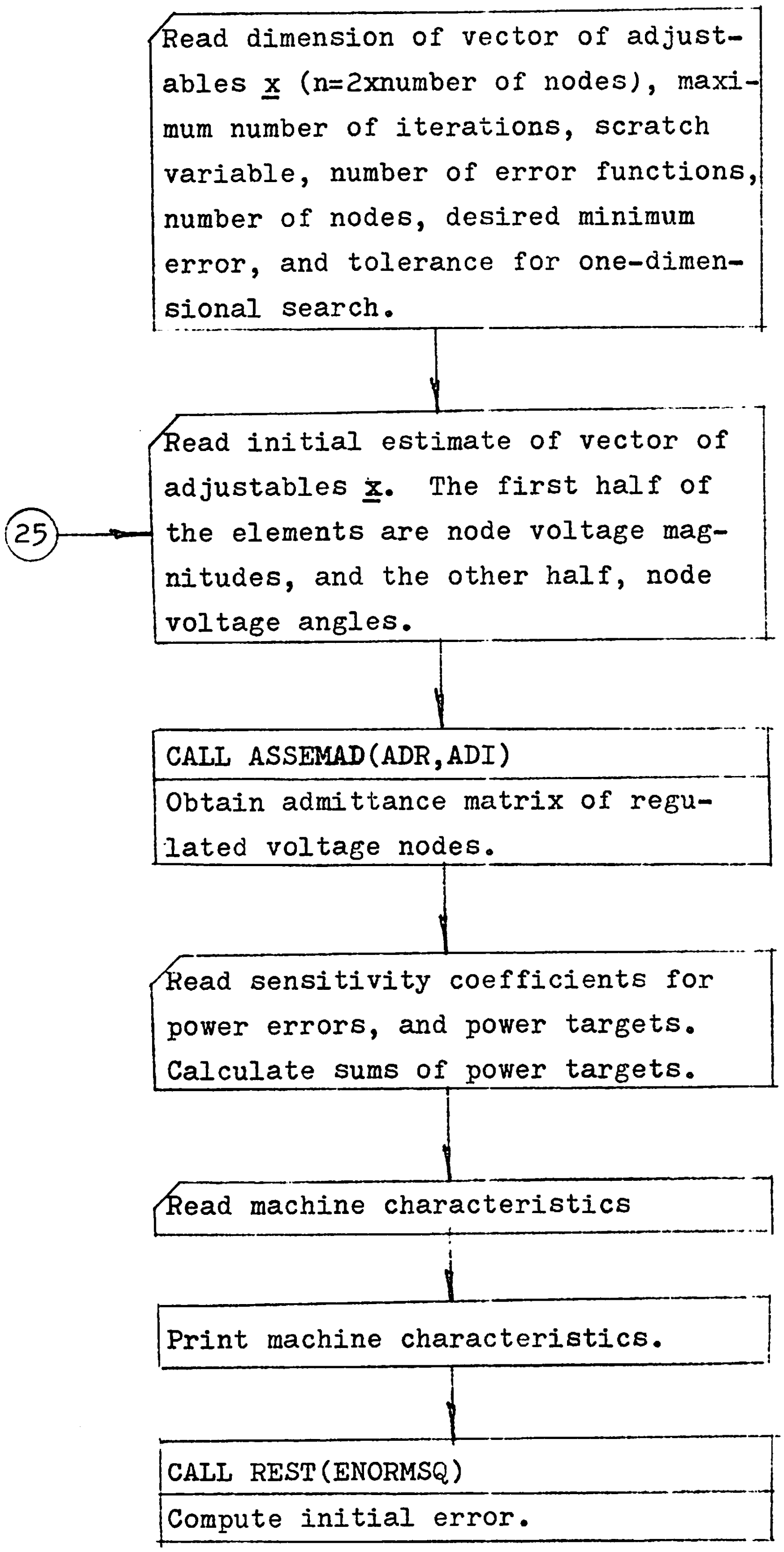
SUBROUTINE NEWTON(ICOUNT)

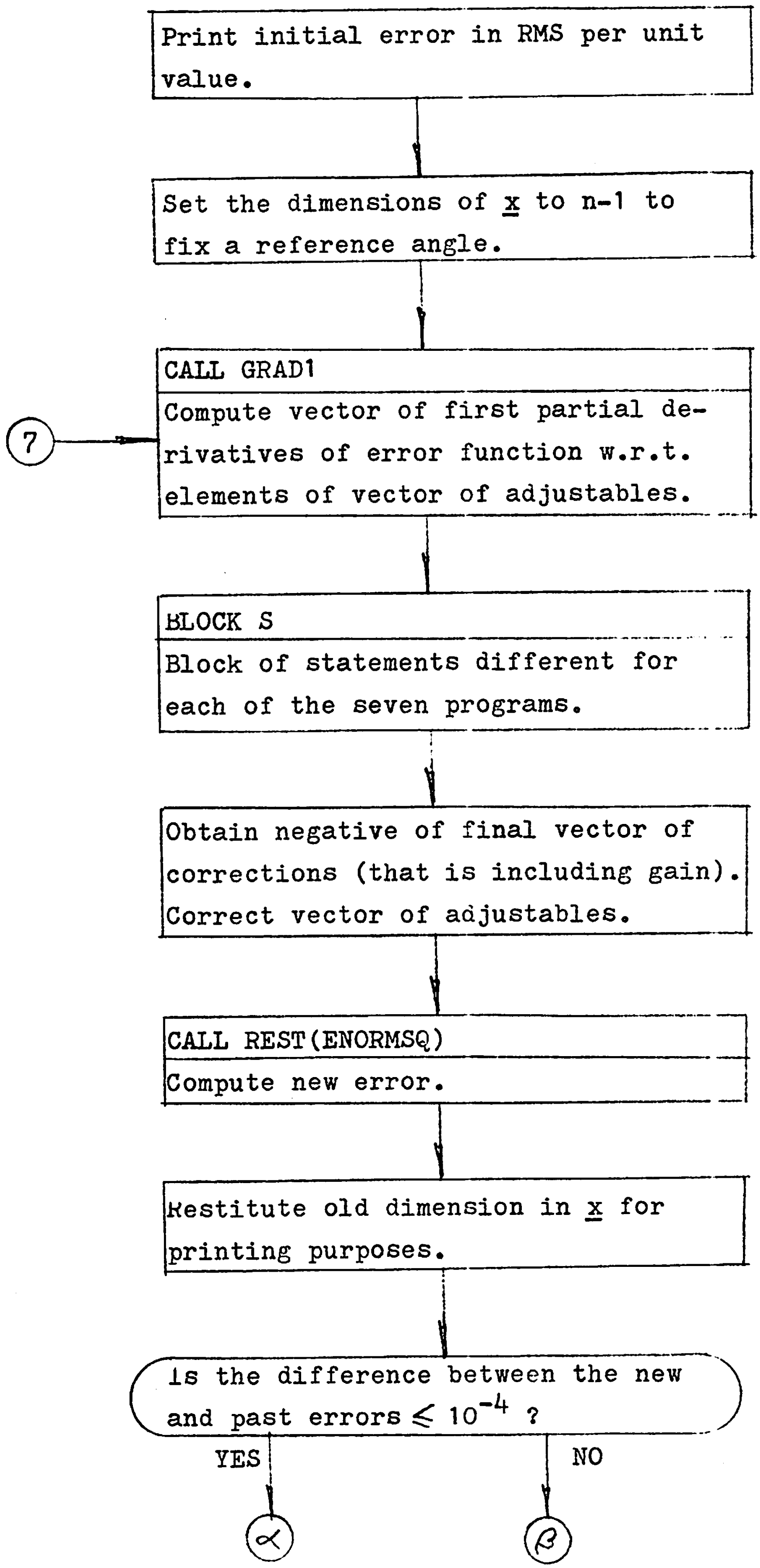


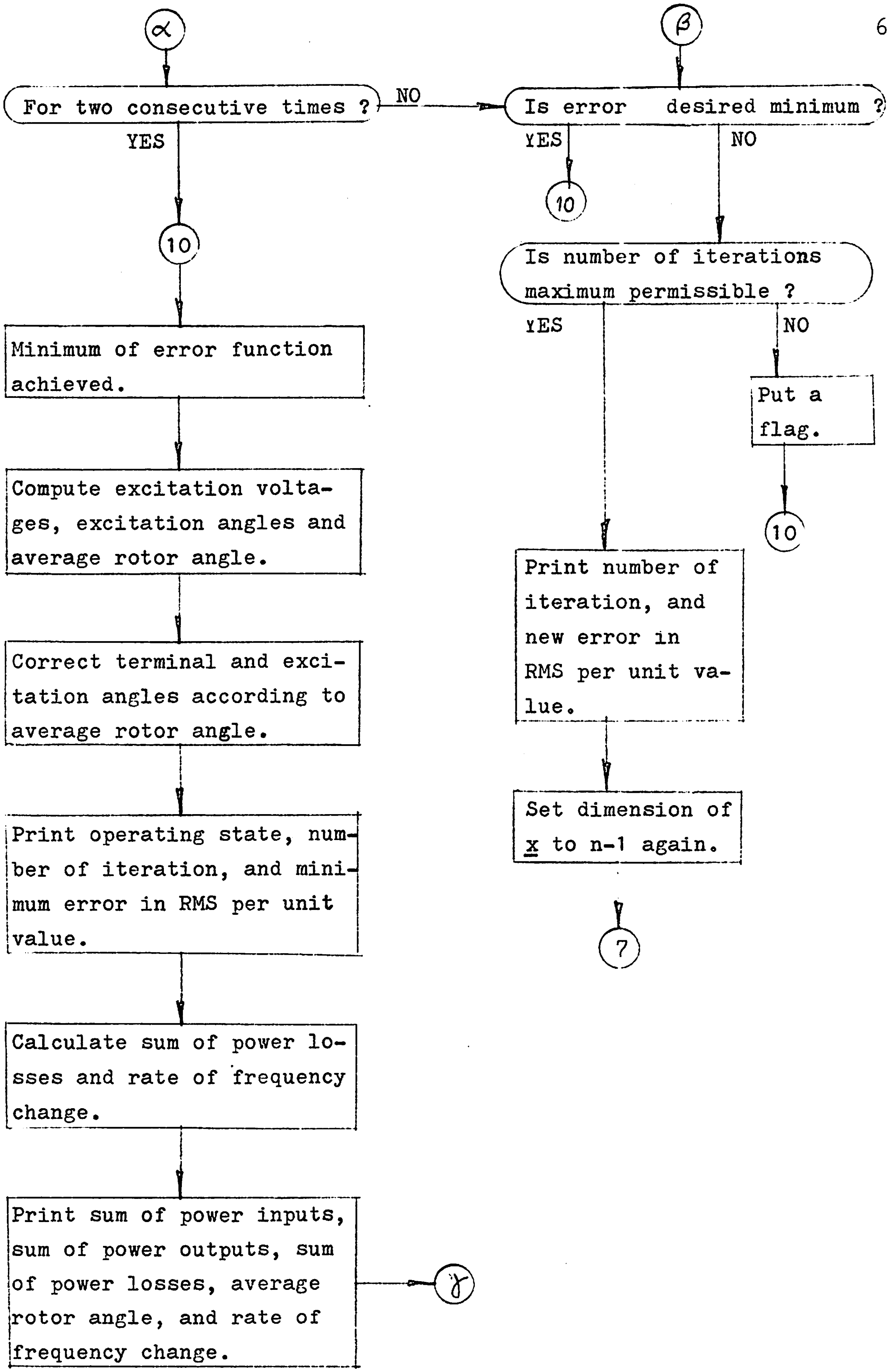
```
      SUBROUTINE NEWTON(ICOUNT)
C     ******************************************************************
C     *THIS SUBROUTINE FINDS THE CORRECTIONS FOR VECTOR X USING NEWTONS  *
C     *METHOD. AT ODD ITERATIONS CORRECTIONS FOR ANGLES ARE CALCULATED   *
C     *ONLY, AT EVEN ITERATIONS CORRECTIONS FOR THE ENTIRE STATE VECTOR  *
C     *ARE FOUND. THE OUTPUT IS THE NEGATIVE OF THE CORRECTIONS STORED IN*
C     *DELX.                                                             *
C     ******************************************************************
      COMMON/SHARE/X(100),DEL(100),A(100,100),N
      COMMON/CRST/DELX(100),DELX0(100)
      COMMON/ADMIT/RNR(8,8),RNI(8,8),MD
      COMMON/DERIV/DPDV(8,8),DPDA(8,8),DQDV(8,8),DQDA(8,8)
      COMMON/ERROR/PE(8),QE(8)
      DIMENSION BB(16,1),IPIV(16),IND(16,2)
      II=ICOUNT/2
      II=II*2
      MD1=MD+1
      IF(ICOUNT.GT.II)GO TO 1
      PRINT 2001
 2001 FORMAT(* NEWTONS METHOD TO CORRECT ENTIRE STATE VECTOR FOLLOWS*/)
      DO 2 I=1,MD
      DO 2 J=1,MD
 2    A(I,J)=DQDV(I,J)
      DO 3 I=1,MD
      DO 3 J=MD1,N
      JA=J-MD
 3    A(I,J)=DQDA(I,JA)
      DO 4 I=MD1,N
      IA=I-MD
      DO 4 J=1,MD
 4    A(I,J)=DPDV(IA,J)
      DO 5 I=MD1,N
      IA=I-MD
      DO 5 J=MD1,N
      JA=J-MD
 5    A(I,J)=DPDA(IA,JA)
      DO 8001 I=1,N
 8001 BB(I,1)=1.0
      CALL MTINV1(BB,0,D,IRROR,IPIV,IND)
      DO 6 I=1,MD
 6    DEL(I)=QE(I)
      DO 7 I=MD1,N
      IA=I-MD
 7    DEL(I)=PE(IA)
      DO 8 I=1,N
      DELX(I)=0.
      DO 8 J=1,N
 8    DELX(I)=DELX(I)+A(I,J)*DEL(J)
      RETURN
 1    MD2=MD-1
      PRINT 2002
 2002 FORMAT(/)
      PRINT 2000
 2000 FORMAT(* NEWTONS METHOD TO CORRECT ANGLES ONLY FOLLOWS*/)
      DO 9 I=1,MD2
```

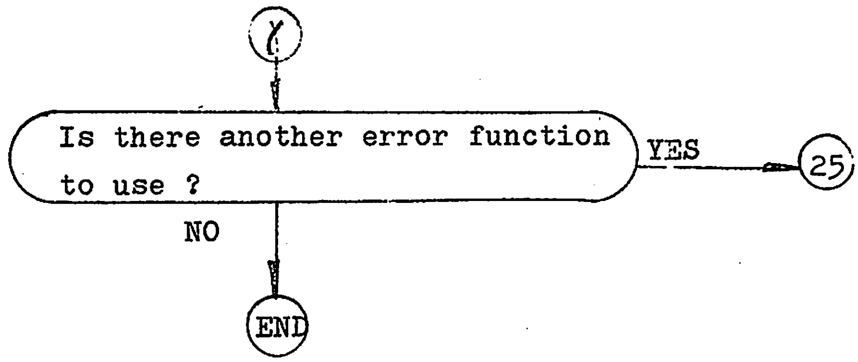
```
      DO 9 J=1,MD2
9     A(I,J)=DPDA(I,J)
      N1=N
      N=MD2
      DO 8002 I=1,N
8002  BB(I,1)=1.0
      CALL MTINV1(BB,0,D,IRROR,IPIV,IND)
      N=N1
      DO 10 I=1,MD2
      IA=I+MD
      DELX(IA)=0.
      DO 10 J=1,MD2
10    DELX(IA)=DELX(IA)+A(I,J)*PE(J)
      DO 11 I=1,MD
11    DELX(I)=0.
      RETURN
      END
```

PROGRAMS NNEWT1, 2, 3, 4, 5, 6, 7




Print initial error in RMS per unit value.
Set the dimensions of x to n-1 to fix a reference angle.
CALL GRAD1
Compute vector of first partial derivatives of error function w.r.t. elements of vector of adjustables.
7
BLOCK S
Block of statements different for each of the seven programs.
Obtain negative of final vector of corrections (that is including gain). Correct vector of adjustables.
CALL REST(ENORMSQ)
Compute new error.
Restitute old dimension in x for printing purposes.
Is the difference between the new and past errors ≤ 10^-4 ?
YES
NO
α
β


α
For two consecutive times ?
NO
YES
10
Minimum of error function achieved.
Compute excitation voltages, excitation angles and average rotor angle.
Correct terminal and excitation angles according to average rotor angle.
Print operating state, number of iteration, and minimum error in RMS per unit value.
Calculate sum of power losses and rate of frequency change.
Print sum of power inputs, sum of power outputs, sum of power losses, average rotor angle, and rate of frequency change.
γ
β
Is error   desired minimum ?
YES
NO
10
Is number of iterations maximum permissible ?
YES
NO
Put a flag.
10
Print number of iteration, and new error in RMS per unit value.
Set dimension of x to n-1 again.
7


γ
Is there another error function to use ?
YES
25
NO
END

PROGRAM NNEWT1

BLOCK S

CALL NEWTON(ICOUNT)
Use Newton's method to obtain negative of vector of corrections.

CALL INTER(AA,BB) CALL ODS(AA,BB,TOLER,100, ALAMB,FMAX)
Select initial interval and perform the one-dimensional search to obtain the optimal to use with the vector of corrections.

```
 2003 FORMAT(4F14.9)
      VATAR(J)=SQRT(PTAR(J)**2+QTAR(J)**2)
      CP2(J)=CP(J)**2
      CQ2(J)=CQ(J)**2
 2002 CONTINUE
      CALL SECOND(PT1)
      PRINT 1
 1    FORMAT(1H1)
      PRINT 160
160   FORMAT(//45H ADMITTANCE MATRIX RN REGULATED VOLTAGE NODES   //)
      DO 164 KA=1,MD,4
      IF(KA+3-MD)162,161,161
161   KB=MD
      GO TO 163
162   KB=KA+3
163   PRINT 313,(JCH,JCH=KA,KB)
313   FORMAT(/4I30/)
      DO 164 J=1,MD
164   PRINT 316,J,(RNR(J,K),RNI(J,K),K=KA,KB)
316   FORMAT(I5,4(F17.2,F13.2))
      PRINT 4008
      PRINT 336
 336  FORMAT(20X,* MACHINE CHARACTERISTICS*/)
      PRINT 337
 337  FORMAT( 66H NODE          INERTIA       RESISTANCE       REACTANCE  MA
     1XIMUM POWER     )
      JCOUNT=0
      SUMIN=0.
C     ******************************************************************
C     *READ MACHINE CHARACTERISTICS, INERTIA, REACTANCE, MAXIMUM POWER,  *
C     *AND NUMBER OF MACHINES.                                           *
C     ******************************************************************
 45   READ 338,J,ROIN(J),ZMR(J),ZMI(J),PSPIN(J),NM
 338  FORMAT(I3,2X,4F15.6,10X,I5)
      SUMIN=SUMIN+ROIN(J)
      JCOUNT=JCOUNT+1
      PRINT 338,J,ROIN(J),ZMR(J),ZMI(J),PSPIN(J)
      IF(JCOUNT.LT.NM)GO TO 45
      PRINT 3000
 3000 FORMAT(///,53X,* INITIAL STATE*/)
C     ******************************************************************
C     *OBTAIN INITIAL ERROR.                                             *
C     ******************************************************************
      CALL REST(ENORMSQ)
      ENA=2.*MD
      EPU=SQRT(ENORMSQ/ENA)
      PRINT 3001
 3001 FORMAT(* NODE   TERM VOLTS   TERM ANG       CP             CQ
     1 POWERS        VARS          VAMPS        PTAR         QTAR        VATA
     1R*)
C     ******************************************************************
C     *PRINT INITIAL STATE.                                              *
C     ******************************************************************
      DO 3002 J=1,MD
 3002 PRINT 56,J,VREF(J),ANGTER(J),CP(J),CQ(J),POWERS(J),VARS(J),VAMPS(J
```

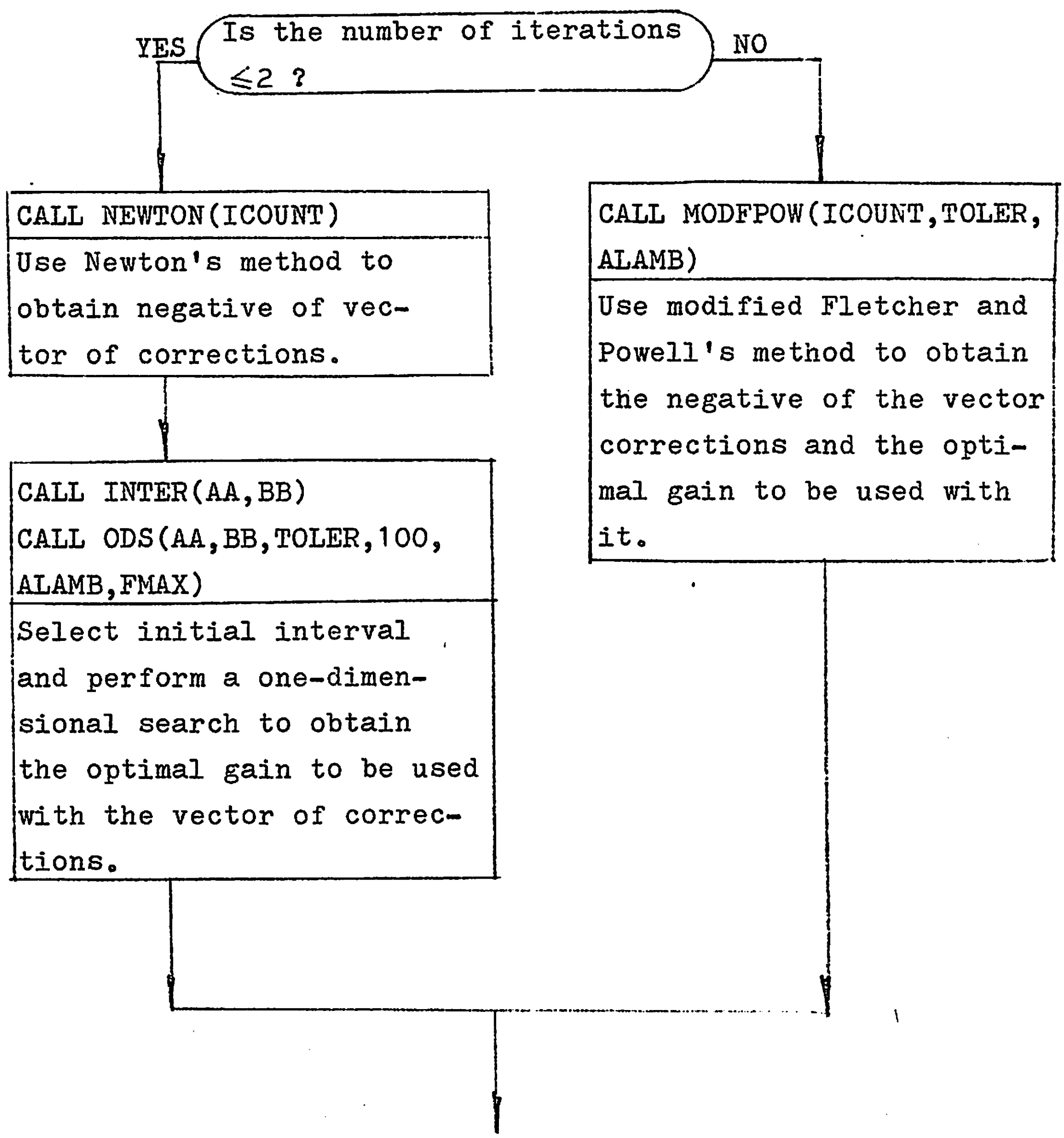
```
     1),PTAR(J),QTAR(J),VATAR(J)
 56   FORMAT(I3,4X,F10.6,2X,F10.6,2X,F10.6,2X,F10.6,2X,F10.4,2X,F10.4,2X
     1,F10.4,2X,F10.4,2X,F10.4,2X,F10.4)
      PRINT 3003
 3003 FORMAT(/)
C     ******************************************************************
C     *PRINT INITIAL ERROR IN RMS PER UNIT VALUE.                      *
C     ******************************************************************
      PRINT 5,EPU
      CALL SECOND(PT2)
      ICOUNT=0
      IE=0
      EPUS=0.
C     ******************************************************************
C     *TAKE ONLY (2*MD-1) ELEMENTS IN X BECAUSE OF REFERENCE ANGLE.    *
C     ******************************************************************
      N=N-1
      CALL SECOND(T2)
      TIME=T2-T1-PT2+PT1
      PRINT 31,TIME
      PRINT 4008
 7    CALL SECOND(T1)
C     ******************************************************************
C     *OBTAIN THE GRADIENT OF ERROR FUNCTION.                          *
C     ******************************************************************
      CALL GRAD1
      ICOUNT=ICOUNT+1
C     ******************************************************************
C     *USE NEWTONS METHOD ONLY.                                        *
C     ******************************************************************
 100  CALL NEWTON(ICOUNT)
C     ******************************************************************
C     *PERFORM A ONE-DIMENSIONAL SEARCH TO OBTAIN OPTIMAL LAMBDA.      *
C     ******************************************************************
      CALL INTER(AA,BB)
      CALL ODS(AA,BB,TOLER,100,ALAMB,FMAX)
C     ******************************************************************
C     *OBTAIN NEGATIVE OF CORRECTIONS AND CORRECT VECTOR X.            *
C     ******************************************************************
      DO 20 J=1,N
      D=ALAMB*DELX(J)
      X(J)=X(J)-D
 20   CONTINUE
C     ******************************************************************
C     *COMPUTE NEW ERROR.                                              *
C     ******************************************************************
      CALL REST(ENORMSQ)
      ENA=2.*MD
      EPU=SQRT(ENORMSQ/ENA)
C     ******************************************************************
C     *RESTITUTE ENTIRE DIMENSION IN X FOR PRINTING PURPOSES.          *
C     ******************************************************************
      N=NN
C     ******************************************************************
C     *CHECK IF THE DIFFERENCE BETWEEN NEW AND PAST ERROR IS LESS THAN *
```

```
C     *OR EQUAL TO E-10 FOR TWO CONSECUTIVE TIMES.                        *
C     ********************************************************************
       DEPU=ABS(EPU-EPUS)
       IF(DEPU.LE.1.0E-10)11,13
 11    IE=IE+1
       IF(IE.GE.2)GO TO 10
       IF(IE.EQ.1)GO TO 12
 13    IE=0
 12    EPUS=EPU
C     ********************************************************************
C     *CHECK IF THE VALUE OF THE ERROR IS LESS THAN OR EQUAL TO THE DESI-*
C     *RED MINIMUM.                                                       *
C     ********************************************************************
       IF(ABS(EPU).LE.EPSI)GO TO 10
C     ********************************************************************
C     *CHECK NUMBER OF ITERATIONS.                                        *
C     ********************************************************************
       M=M-1
       IF(M.LE.0) GO TO 35
       CALL SECOND(T2)
C     ********************************************************************
C     *PRINT NEW RMS PER UNIT ERROR.                                      *
C     ********************************************************************
       PRINT 9,ICOUNT
       PRINT 5,EPU
       TIME=TIME+T2-T1
       PRINT 30,TIME
       PRINT 4008
 4008  FORMAT(//)
C     ********************************************************************
C     *SET DIMENSION OF X TO N-1 AND PERFORM ANOTHER ITERATION.           *
C     ********************************************************************
       N=N-1
       GO TO 7
   35  PRINT 6, ALAMB
    6  FORMAT(* MORE ITERATIONS NEEDED LAMBDA=*,F10.4)
   10  CONTINUE
       CALL SECOND(T2)
       TIME=TIME+T2-T1
       AVEANG=0.
C     ********************************************************************
C     *COMPUTE EXCITATION VOLTAGES AND EXCITATION ANGLES.                 *
C     ********************************************************************
       DO 52 J=1,NM
       VMR=ZMR(J)*UMR(J)-ZMI(J)*UMI(J)+VREF(J)*COS(ANGTER(J))
       VMI=ZMI(J)*UMR(J)+ZMR(J)*UMI(J)+VREF(J)*SIN(ANGTER(J))
       VM(J)=SQRT(VMR**2+VMI**2)
       ANGMACH(J)=ATAN2(VMI,VMR)
C     ********************************************************************
C     *CALCULATE AVERAGE ANGLE.                                           *
C     ********************************************************************
 52    AVEANG=AVEANG+ROIN(J)*ANGMACH(J)/SUMIN
C     ********************************************************************
C     *CORRECT TERMINAL AND EXCITATION ANGLES.                            *
C     ********************************************************************
```

```
      DO 59 J=1,MD
 59   ANGTER(J)=ANGTER(J)-AVEANG+0.3
      DO 60 J=1,NM
 60   ANGMACH(J)=ANGMACH(J)-AVEANG+0.3
      PRINT 3006
 3006 FORMAT(53X,* OPERATING STATE*/)
      PRINT 53
 53   FORMAT(* NODE   TERM VOLTS   TERM ANG   EXC VOLTS    EXC ANGLE
     1 POWERS        VARS          VAMPS         PTAR           QTAR          VATA
     1R*)
C     ******************************************************************
C     *PRINT OPERATING STATE.                                          *
C     ******************************************************************
      DO 54 J=1,MD
      PRINT 56,J,VREF(J),ANGTER(J),VM(J),ANGMACH(J),POWERS(J),VARS(J),VA
     1MPS(J),PTAR(J),QTAR(J),VATAR(J)
 54   CONTINUE
      PRINT 3003
      PRINT 9,ICOUNT
      PRINT 2005,EPU
      PRINT 30,TIME
C     ******************************************************************
C     *CALCULATE LOSSES.                                               *
C     ******************************************************************
      DO 70 J=1,MD
      DO 70 K=1,MD
 70   TOTALPO=TOTALPO+RNR(K,J)*VREF(J)**2
      SUMLOSS=TOTALPI-TOTALPO
      PRINT 57,AVEANG
 57   FORMAT(* AVERAGE ROTOR ANGLE =*,F10.6)
      PRINT 66,TOTALPI
 66   FORMAT(* SUM OF POWER INPUTS =*,F15.6)
      PRINT 67,TOTALPO
 67   FORMAT(* SUM OF POWER OUTPUTS =*,F15.6)
      PRINT 58,SUMLOSS
 58   FORMAT(* SUM OF POWER LOSSES =*,F15.6)
C     ******************************************************************
C     *CHECK IF THERE IS ANOTHER ERROR FUNCTION TO USE.                *
C     ******************************************************************
      IF(INDEX.LT.NF) GO TO 25
 5    FORMAT(* EPU =*,F10.5)
 2005 FORMAT(* MIN EPU =*,F10.5)
 9    FORMAT(* ITERATION NUMBER*,I3)
 30   FORMAT(* TIME AT THIS ITERATION =*,F10.5)
 31   FORMAT(* TIME SPENT BEFORE ITERATING =*,F10.5)
      RETURN
      END
```

PROGRAM NNEWT2

BLOCK S



```
J7512CS,7,060,70000.7512 ZARATE NNEWT2
RUN,S.
LGO.
'
      PROGRAM NNEWT2(INPUT,OUTPUT)
C     ****************************************************************
C     *THIS PROGRAM OBTAINS THE OPERATING STATE OF A POWER SYSTEM FROM AN*
C     *INITIAL ESTIMATED STATE. THE PROGRAM USES NEWTONS METHOD FOR THE  *
C     *FIRST TWO ITERATIONS AND THEN A MODIFICATION TO THE FLETCHER AND  *
C     *POWELLS METHOD, UNTIL SATISFACTORY MINIMIZATION OF ERROR FUNCTION *
C     *IS ACCOMPLISHED.                                                  *
C     ****************************************************************
      COMMON INDEX
      COMMON/FLET/G(20),S(20),AC(20,20),BC(20,20),ICOUNTF
      COMMON/SHARE/X(100),DEL(100),A(100,100),N
      COMMON/CRST/DELX(100),DELX0(100)
      COMMON/ADMIT/RNR(8,8),RNI(8,8),MD
      COMMON/COSTS/CP2(8),CQ2(8)
      COMMON/TAR/PTAR(8),QTAR(8),VATAR(8)
      COMMON/DERIV/DPDV(8,8),DPDA(8,8),DQDV(8,8),DQDA(8,8)
      COMMON/ERROR/PE(8),QE(8)
      COMMON/REF/VREF(8),ANGTER(8),VR(8),VI(8),UMR(8),UMI(8)
      COMMON/POWER/TOTALPO,TOTALPI,POWERS(8),VARS(8),VAMPS(8)
      COMMON/SUMS/TOTALP,TOTALQ,TOTALVA
      DIMENSION ROIN(8),ZMR(8),ZMI(8),PSPIN(8),VM(8),ANGMACH(8)
      DIMENSION CP(8),CQ(8)
      DIMENSION ADR(8,8),ADI(8,8)
      CALL SECOND(T1)
C     ****************************************************************
C     *READ DIMENSION OF VECTOR X (N=2*MD), MAXIMUM NUMBER OF ITERATIONS,*
C     *SCRATCH VARIABLE, NUMBER OF ERROR FUNCTIONS TO USE, NUMBER OF NO- *
C     *DES, TOLERANCE FOR MINIMUM VALUE OF ERROR FUNCTION, AND TOLERANCE *
C     *FOR ONE-DIMENSIONAL SEARCH.                                       *
C     ****************************************************************
      READ 3,N,M,K,NF,MD,EPSI,TOLER
 3    FORMAT(5I5,2F10.6)
      NN=N
      INDEX=0
   25 INDEX=INDEX+1
C     ****************************************************************
C     *READ INITIAL ESTIMATE OF NODE VOLTAGES AND ANGLES GIVEN AS A VEC- *
C     *TOR X. THE FIRST MD ELEMENTS OF THIS VECTOR ARE THE NODE VOLTAGES,*
C     *AND THE LAST MD ELEMENTS ARE ANGLES. DIMENSION OF X IS THEN N.    *
C     ****************************************************************
      READ 2,(X(I),I=1,N)
    2 FORMAT(8F10.6)
      CALL SECOND(PT1)
      PRINT 1
 1    FORMAT(1H1)
C     ****************************************************************
C     *OBTAIN ADMITTANCE MATRIX OF REGULATED VOLTAGE NODES               *
C     ****************************************************************
      CALL ASSEMAD(ADR,ADI)
      DO 2000 J=1,MD
      DO 2000 I=1,MD
```

```
      RNR(I,J)=ADR(I,J)
 2000 RNI(I,J)=ADI(I,J)
C     ******************************************************************
C     *READ SENSITIVITY COEFFICIENTS, AND POWER TARGETS.                *
C     ******************************************************************
      SUMP=0.
      SUMQ=0.
      SUMVA=0.
      DO 2002 J=1,MD
      READ 2003,CP(J),CQ(J),PTAR(J),QTAR(J)
 2003 FORMAT(4F14.9)
      VATAR(J)=SQRT(PTAR(J)**2+QTAR(J)**2)
      SUMP=SUMP+PTAR(J)
      SUMQ=SUMQ+QTAR(J)
      SUMVA=SUMVA+VATAR(J)
      CP2(J)=CP(J)**2
      CQ2(J)=CQ(J)**2
 2002 CONTINUE
      PRINT 4008
      PRINT 336
 336  FORMAT(20X,* MACHINE CHARACTERISTICS*/)
      PRINT 337
 337  FORMAT( 66H NODE        INERTIA      RESISTANCE      REACTANCE  MA
     1XIMUM POWER     )
      JCOUNT=0
      SUMIN=0.
C     ******************************************************************
C     *READ MACHINE CHARACTERISTICS, INERTIA, REACTANCE, MAXIMUM POWER, *
C     *AND NUMBER OF MACHINES.                                          *
C     ******************************************************************
 45   READ 338,J,ROIN(J),ZMR(J),ZMI(J),PSPIN(J),NM
 338  FORMAT(I3,2X,4F15.6,10X,I5)
      SUMIN=SUMIN+ROIN(J)
      JCOUNT=JCOUNT+1
      PSPIN(J)=PSPIN(J)/100.
      PRINT 338,J,ROIN(J),ZMR(J),ZMI(J),PSPIN(J)
      IF(JCOUNT.LT.NM)GO TO 45
      PRINT 3000
 3000 FORMAT(///,53X,* INITIAL STATE*/)
C     ******************************************************************
C     *OBTAIN INITIAL ERROR.                                            *
C     ******************************************************************
      CALL REST(ENORMSQ)
      ENA=2.*MD
      EPU=SQRT(ENORMSQ/ENA)
      PRINT 531
 531  FORMAT(* NODE     TERMINAL VOLTAGES      POWER ERROR WEIGHTS
     1        COMPUTED POWERS                        POWER TARGETS*)
      PRINT 3001
 3001 FORMAT(*        TERM VOLTS   TERM ANG      CP            CQ
     1 POWERS        VARS          VAMPS        PTAR          QTAR          VATA
     1R*)
C     ******************************************************************
C     *PRINT INITIAL STATE.                                             *
C     ******************************************************************
```

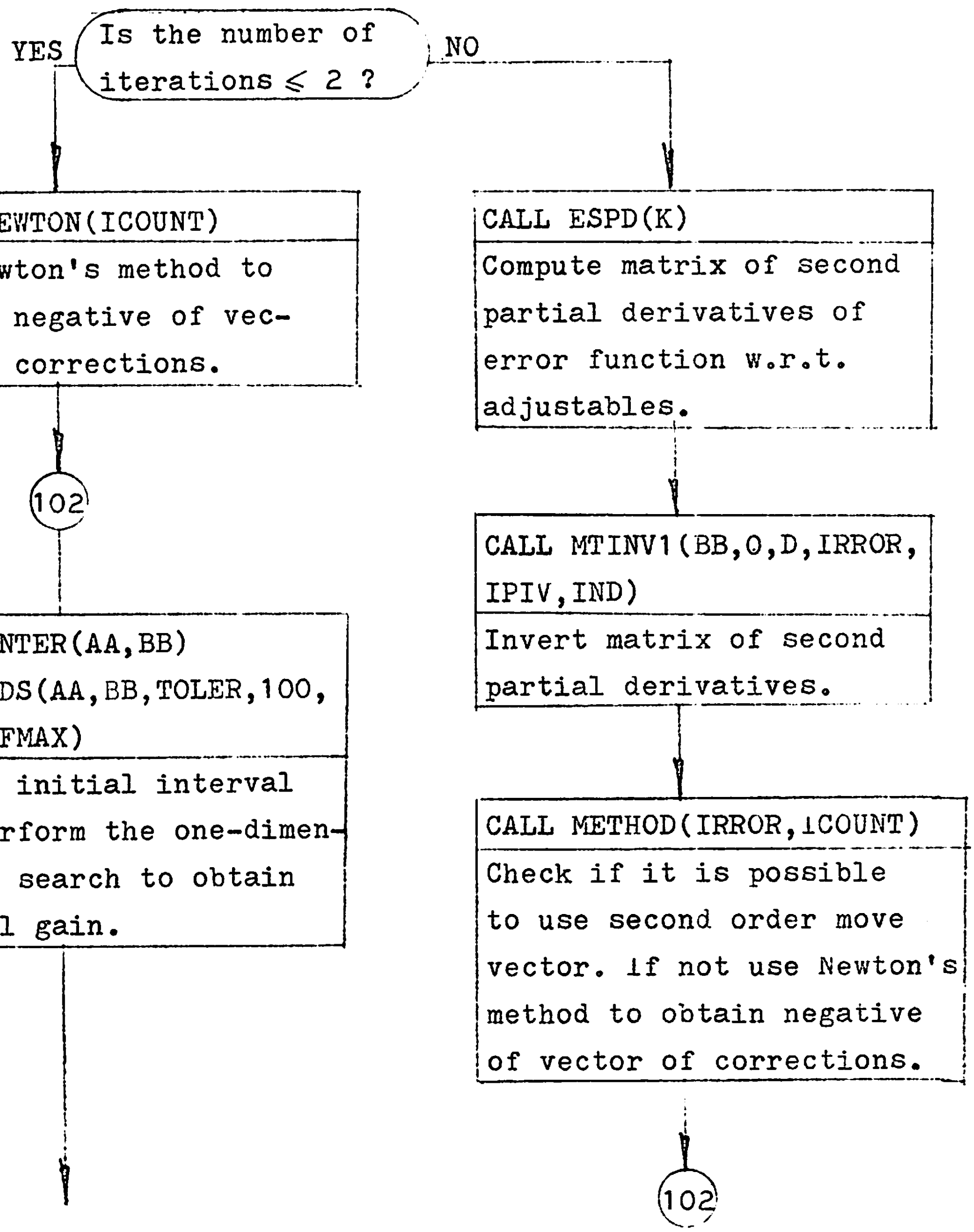
```
      DO 3002 J=1,MD
 3002 PRINT 56,J,VREF(J),ANGTER(J),CP(J),CQ(J),POWERS(J),VARS(J),VAMPS(J
     1),PTAR(J),QTAR(J),VATAR(J)
 56   FORMAT(I3,4X,F10.6,2X,F10.6,2X,F10.6,2X,F10.6,2X,F10.4,2X,F10.4,2X
     1,F10.4,2X,F10.4,2X,F10.4,2X,F10.4)
      PRINT 561,TOTALP,TOTALQ,TOTALVA,SUMP,SUMQ,SUMVA
 561  FORMAT(* TOTALS*,48X,F10.4,2X,F10.4,2X,F10.4,2X,F10.4,2X,F10.4,2X,
     1F10.4)
      PRINT 3003
 3003 FORMAT(/)
C     ******************************************************************
C     *PRINT INITIAL ERROR IN RMS PER UNIT VALUE.                      *
C     ******************************************************************
      PRINT 5,EPU
      CALL SECOND(PT2)
      ICOUNT=0
      IE=0
      EPUS=0.
      EPUSP=0.
C     ******************************************************************
C     *TAKE ONLY (2*MD-1) ELEMENTS IN X BECAUSE OF REFERENCE ANGLE.    *
C     ******************************************************************
      N=N-1
      CALL SECOND(T2)
      TIME=T2-T1-PT2+PT1
      PRINT 31,TIME
      PRINT 4008
      ICOUNTF=0
 7    CALL SECOND(T1)
C     ******************************************************************
C     *OBTAIN THE GRADIENT OF ERROR FUNCTION.                          *
C     ******************************************************************
      CALL GRAD1
      ICOUNT=ICOUNT+1
      IF(ICOUNT.LE.2)100,101
C     ******************************************************************
C     *USE NEWTONS METHOD FOR FIRST TWO ITERATIONS.                    *
C     ******************************************************************
 100  CALL NEWTON(ICOUNT)
C     ******************************************************************
C     *PERFORM A ONE-DIMENSIONAL SEARCH TO OBTAIN OPTIMAL LAMBDA.      *
C     ******************************************************************
      CALL INTER(AA,BB)
      CALL ODS(AA,BB,TOLER,100,ALAMB,FMAX)
      GO TO 261
C     ******************************************************************
C     *USE MODIFIED FLETCHER AND POWELLS METHOD FOR THE REMAINING ITERA- *
C     *TIONS.                                                          *
C     ******************************************************************
 101  CONTINUE
      CALL MODFPOW(ICOUNT,TOLER,ALAMB)
 261  CONTINUE
      DO 20 J=1,N
C     ******************************************************************
C     *OBTAIN NEGATIVE OF CORRECTIONS AND CORRECT VECTOR X.            *
```

```
    6 FORMAT(* MORE ITERATIONS NEEDED LAMBDA=*,F10.4)
   10 CONTINUE
      CALL SECOND(T2)
      TIME=TIME+T2-T1
      AVEANG=0.
C     *****************************************************************
C     *COMPUTE EXCITATION VOLTAGES AND EXCITATION ANGLES.             *
C     *****************************************************************
      DO 52 J=1,NM
      VMR=ZMR(J)*UMR(J)-ZMI(J)*UMI(J)+VREF(J)*COS(ANGTER(J))
      VMI=ZMI(J)*UMR(J)+ZMR(J)*UMI(J)+VREF(J)*SIN(ANGTER(J))
      VM(J)=SQRT(VMR**2+VMI**2)
      ANGMACH(J)=ATAN2(VMI,VMR)
C     *****************************************************************
C     *CALCULATE AVERAGE ANGLE.                                        *
C     *****************************************************************
   52 AVEANG=AVEANG+ROIN(J)*ANGMACH(J)/SUMIN
C     *****************************************************************
C     *CORRECT TERMINAL AND EXCITATION ANGLES.                         *
C     *****************************************************************
      DO 59 J=1,MD
   59 ANGTER(J)=ANGTER(J)-AVEANG+0.3
      DO 60 J=1,NM
   60 ANGMACH(J)=ANGMACH(J)-AVEANG+0.3
      AVEANG=0.3
      RATEFQ=50.0*(TOTALP-SUMP)/SUMIN
      PRINT 3006
 3006 FORMAT(53X,* OPERATING STATE*/)
      PRINT 532
  532 FORMAT(* NODE      TERMINAL VOLTAGES         GENERATOR VOLTAGES
     1        COMPUTED POWERS                          POWER TARGETS*)
      PRINT 53
   53 FORMAT(*           TERM VOLTS    TERM ANG    EXC VOLTS     EXC ANGLE
     1 POWERS          VARS            VAMPS          PTAR            QTAR             VATA
     1R*)
C     *****************************************************************
C     *PRINT OPERATING STATE.                                          *
C     *****************************************************************
      DO 54 J=1,MD
      PRINT 56,J,VREF(J),ANGTER(J),VM(J),ANGMACH(J),POWERS(J),VARS(J),VA
     1MPS(J),PTAR(J),QTAR(J),VATAR(J)
   54 CONTINUE
      PRINT 561,TOTALP,TOTALQ,TOTALVA,SUMP,SUMQ,SUMVA
      PRINT 3003
      PRINT 9,ICOUNT
      PRINT 2005,EPU
      PRINT 30,TIME
C     *****************************************************************
C     *CALCULATE LOSSES.                                               *
C     *****************************************************************
      DO 70 J=1,MD
      DO 70 K=1,MD
   70 TOTALPO=TOTALPO+RNR(K,J)*VREF(J)**2
      SUMLOSS=TOTALPI-TOTALPO
      PRINT 57,AVEANG
```

```
C      DQDA      PARTIAL DERIVATIVE OF REACTIVE POWER WITH RESPECT TO AN-
C                GLE.
C      DQDV      PARTIAL DERIVATIVE OF REACTIVE POWER WITH RESPECT TO VOL-
C                TAGE.
C      ENA       TWICE THE NUMBER OF REGULATED VOLTAGE NODES.
C      ENORMSQ   ERROR FUNCTION USED BY THE MINIMIZATION TECHNIQUES.
C      EPSI      TOLERANCE FOR MINIMUM OF ERROR FUNCTION.
C      EPU       VALUE OF ERROR FUNCTION IN RMS PER UNIT.
C      EPUS      SCRATCH VARIABLE TO STORE VALUE OF EPU.
C      ETA       PRELIMINARY GAIN IN SUBROUTINES MODFPOW AND FPOWELL.
C      F         SENDING-END VOLTAGES RECIPROCAL ADMITTANCE MATRIX L BY M.
C      FAL       VALUE OF ERROR FUNCTION ASSOCIATED WITH GAIN AL IN SUBROU
C                TINE FUNCT.
C      G         RECEIVING END VOLTAGES RECIPROCAL ADMITTANCE MATRIX L BY
C                M. SCRATCH VARIABLE IN MODFPOW AND FPOWELL.
C      H         F-G
C      ICOUNT    COUNTER OF NUMBER OF ITERATIONS IN MAIN PROGRAM.
C      ICOUNTF   LOCAL COUNTER FOR SUBROUTINES MODFPOW AND FPOWELL.
C      IE        COUNTER OF THE NUMBER OF TIMES EPU HAS NOT CHANGED MORE
C                THAN E-04 .
C      INDEX     COUNTER OF NUMBER OF ERROR FUNCTIONS.
C      IRROR     DETECTOR FOR SINGULARITY OF MATRICES.
C      JCOUNT    COUNTER OF NUMBER OF CARDS READ AS MACHINE CHARACTERIS-
C                TICS.
C      K         SCRATCH VARIABLE TO SKIP THE USE OF SUBROUTINE ESPD.
C      L         NUMBER OF MACHINES WITH VOLTAGE REGULATORS. MAXIMUM NUM-
C                BER OF ITERATIONS IN SUBROUTINE ODS.
C      LC        NUMBER OF CIRCUIT LOST.
C      LN        NUMBER OF LINE LOST.
C      M         MAXIMUM NUMBER OF ITERATIONS IN MAIN PROGRAM. NUMBER OF
C                MACHINES WITH VOLTAGE REGULATORS IN SUBROUTINE ASSEMAD.
C      MD        NUMBER OF REGULATED VOLTAGE NODES.
C      N         DIMENSION OF VECTOR OF ADJUSTABLES.
C      NF        NUMBER OF ERROR FUNCTIONS TO USE.
C      NM        NUMBER OF GENERATORS.
C      NN        SCRATCH VARIABLE.
C      NODES     TOTAL NUMBER OF NODES.
C
C      POWER BASE IS 100 MEGAWATTS.
C
C      PE        ERROR IN REAL POWER.
C      POWER     COMPUTED REAL POWER AT EACH NODE.
C      POWERS    COMPUTED REAL POWER AT EACH NODE.
C      PSPIN     MAXIMUM REAL POWER AVAILABLE FROM EACH GENERATOR.
C      PTAR      REAL POWER TARGET AT EACH NODE.
C      QE        ERROR IN REACTIVE POWER.
C      QTAR      REACTIVE POWER TARGET AT EACH NODE.
C      R         RECIPROCAL ADMITTANCE REGULATED VOLTAGES.
C      RATEFQ    RATE OF CHANGE OF FREQUENCY IN PERCENT PER SECOND.
C      RNI       IMAGINARY PART OF ADMITTANCE BETWEEN TWO REGULATED VOLTA-
C                GE NODES.
C      RNR       REAL PART OF ADMITTANCE BETWEEN TWO REGULATED VOLTAGE NO-
C                DES.
C      ROIN      ROTOR INERTIA. GIVEN IN 100 MEGAWATT-SECOND STORED ENERGY.
C      RZ        REAL PART OF INTERNAL IMPEDANCE.
```

PROGRAM MNNEWT3

BLOCK S



```
J7512CS,7,060,70000.7512 ZARATE NNEWT3
RUN,S.
LGO.
'
      PROGRAM NNEWT3(INPUT,OUTPUT)
C     *****************************************************************
C     *THIS PROGRAM OBTAINS THE OPERATING STATE OF A POWER SYSTEM FROM AN*
C     *INITIAL ESTIMATED STATE. THE PROGRAM USES NEWTONS METHOD FOR THE  *
C     *FIRST TWO ITERATIONS AND THEN THE SECOND ORDER MOVE VECTOR METHOD,*
C     *UNTIL SATISFACTORY MINIMIZATION OF ERROR FUNCTION IS ACCOMPLISHED.*
C     *****************************************************************
      COMMON INDEX
      COMMON/SHARE/X(100),DEL(100),A(100,100),N
      COMMON/CRST/DELX(100),DELXO(100)
      COMMON/ADMIT/RNR(8,8),RNI(8,8),MD
      COMMON/COSTS/CP2(8),CQ2(8)
      COMMON/TAR/PTAR(8),QTAR(8),VATAR(8)
      COMMON/DERIV/DPDV(8,8),DPDA(8,8),DQDV(8,8),DQDA(8,8)
      COMMON/ERROR/PE(8),QE(8)
      COMMON/REF/VREF(8),ANGTER(8),VR(8),VI(8),UMR(8),UMI(8)
      COMMON/POWER/TOTALPO,TOTALPI,POWERS(8),VARS(8),VAMPS(8)
      DIMENSION ROIN(8),ZMR(8),ZMI(8),PSPIN(8),VM(8),ANGMACH(8)
      DIMENSION CP(8),CQ(8)
      DIMENSION BB(16,1),IPIV(16),IND(16,2)
      CALL SECOND(T1)
C     *****************************************************************
C     *READ DIMENSION OF VECTOR X (N=2*MD), MAXIMUM NUMBER OF ITERATIONS,*
C     *SCRATCH VARIABLE, NUMBER OF ERROR FUNCTIONS TO USE, NUMBER OF NO- *
C     *DES, TOLERANCE FOR MINIMUM VALUE OF ERROR FUNCTION, AND TOLERANCE *
C     *FOR ONE-DIMENSIONAL SEARCH.                                       *
C     *****************************************************************
      READ 3,N,M,K,NF,MD,EPSI,TOLER
 3    FORMAT(5I5,2F10.6)
      NN=N
      INDEX=0
   25 INDEX=INDEX+1
C     *****************************************************************
C     *READ INITIAL ESTIMATE OF NODE VOLTAGES AND ANGLES GIVEN AS A VEC- *
C     *TOR X. THE FIRST MD ELEMENTS OF THIS VECTOR ARE THE NODE VOLTAGES,*
C     *AND THE LAST MD ELEMENTS ARE ANGLES. DIMENSION OF X IS THEN N.    *
C     *****************************************************************
      READ 2,(X(I),I=1,N)
    2 FORMAT(8F10.6)
C     *****************************************************************
C     *READ ADMITTANCE MATRIX OF REGULATED VOLTAGE NODES.                *
C     *****************************************************************
      DO 2000 J=1,MD
      DO 2000 I=1,MD
      READ 2001,RNR(I,J),RNI(I,J)
 2001 FORMAT(2F10.6)
 2000 CONTINUE
C     *****************************************************************
C     *READ SENSITIVITY COEFFICIENTS, AND POWER TARGETS.                 *
C     *****************************************************************
      DO 2002 J=1,MD
```

```
      READ 2003,CP(J),CQ(J),PTAR(J),QTAR(J)
 2003 FORMAT(4F14.9)
      VATAR(J)=SQRT(PTAR(J)**2+QTAR(J)**2)
      CP2(J)=CP(J)**2
      CQ2(J)=CQ(J)**2
 2002 CONTINUE
      CALL SECOND(PT1)
      PRINT 1
 1    FORMAT(1H1)
      PRINT 160
160   FORMAT(//45H ADMITTANCE MATRIX RN REGULATED VOLTAGE NODES   //)
      DO 164 KA=1,MD,4
      IF(KA+3-MD)162,161,161
161   KB=MD
      GO TO 163
162   KB=KA+3
163   PRINT 313,(JCH,JCH=KA,KB)
313   FORMAT(/4I30/)
      DO 164 J=1,MD
164   PRINT 316,J,(RNR(J,K),RNI(J,K),K=KA,KB)
316   FORMAT(I5,4(F17.2,F13.2))
      PRINT 4008
      PRINT 336
 336  FORMAT(20X,* MACHINE CHARACTERISTICS*/)
      PRINT 337
 337  FORMAT( 66H NODE         INERTIA       RESISTANCE       REACTANCE  MA
     1XIMUM POWER     )
      JCOUNT=0
      SUMIN=0.
C     *****************************************************************
C     *READ MACHINE CHARACTERISTICS, INERTIA, REACTANCE, MAXIMUM POWER, *
C     *AND NUMBER OF MACHINES.                                          *
C     *****************************************************************
 45   READ 338,J,ROIN(J),ZMR(J),ZMI(J),PSPIN(J),NM
 338  FORMAT(I3,2X,4F15.6,10X,I5)
      SUMIN=SUMIN+ROIN(J)
      JCOUNT=JCOUNT+1
      PRINT 338,J,ROIN(J),ZMR(J),ZMI(J),PSPIN(J)
      IF(JCOUNT.LT.NM)GO TO 45
      PRINT 3000
 3000 FORMAT(///,53X,* INITIAL STATE*/)
C     *****************************************************************
C     *OBTAIN INITIAL ERROR.                                            *
C     *****************************************************************
      CALL REST(ENORMSQ)
      ENA=2.*MD
      EPU=SQRT(ENORMSQ/ENA)
      PRINT 3001
 3001 FORMAT(* NODE    TERM VOLTS   TERM ANG       CP             CQ
     1 POWERS        VARS          VAMPS          PTAR           QTAR          VATA
     1R*)
C     *****************************************************************
C     *PRINT INITIAL STATE.                                             *
C     *****************************************************************
      DO 3002 J=1,MD
```

```
3002 PRINT 56,J,VREF(J),ANGTER(J),CP(J),CQ(J),POWERS(J),VARS(J),VAMPS(J
    1),PTAR(J),QTAR(J),VATAR(J)
 56  FORMAT(I3,4X,F10.6,2X,F10.6,2X,F10.6,2X,F10.6,2X,F10.4,2X,F10.4,2X
    1,F10.4,2X,F10.4,2X,F10.4,2X,F10.4)
     PRINT 3003
3003 FORMAT(/)
C    ******************************************************************
C    *PRINT INITIAL ERROR IN RMS PER UNIT VALUE.                      *
C    ******************************************************************
     PRINT 5,EPU
     CALL SECOND(PT2)
     ICOUNT=0
     IE=0
     EPUS=0.
C    ******************************************************************
C    *TAKE ONLY (2*MD-1) ELEMENTS IN X BECAUSE OF REFERENCE ANGLE.    *
C    ******************************************************************
     N=N-1
     CALL SECOND(T2)
     TIME=T2-T1-PT2+PT1
     PRINT 31,TIME
     PRINT 4008
 7   CALL SECOND(T1)
C    ******************************************************************
C    *OBTAIN THE GRADIENT OF ERROR FUNCTION.                          *
C    ******************************************************************
     CALL GRAD1
     ICOUNT=ICOUNT+1
     IF(ICOUNT.LE.2)100,101
C    ******************************************************************
C    *USE NEWTONS METHOD FOR FIRST TWO ITERATIONS.                    *
C    ******************************************************************
 100 CALL NEWTON(ICOUNT)
     GO TO 102
 101 CONTINUE
C    ******************************************************************
C    *USE SECOND ORDER MOVE VECTOR FOR THE REMAINING ITERATIONS.      *
C    *COMPUTE MATRIX OF SECOND PARTIAL DERIVATIVES OF ERROR FUNCTION  *
C    *WITH RESPECT TO ADJUSTABLES.                                    *
C    ******************************************************************
     CALL ESPD(K)
     DO 8001 I=1,N
8001 BB(I,1)=1.0
C    ******************************************************************
C    *INVERT MATRIX OF SECOND PARTIAL DERIVATIVES.                    *
C    ******************************************************************
     CALL MTINV1(BB,0,D,IRROR,IPIV,IND)
C    ******************************************************************
C    *CHECK IF IT IS POSSIBLE TO USE THE SECOND ORDER MOVE VECTOR. OB- *
C    *TAIN VECTOR OF CORRECTIONS.                                     *
C    ******************************************************************
     CALL METHOD(IRROR,ICOUNT)
 102 CONTINUE
C    ******************************************************************
C    *PERFORM A ONE-DIMENSIONAL SEARCH TO OBTAIN OPTIMAL LAMBDA.      *
```

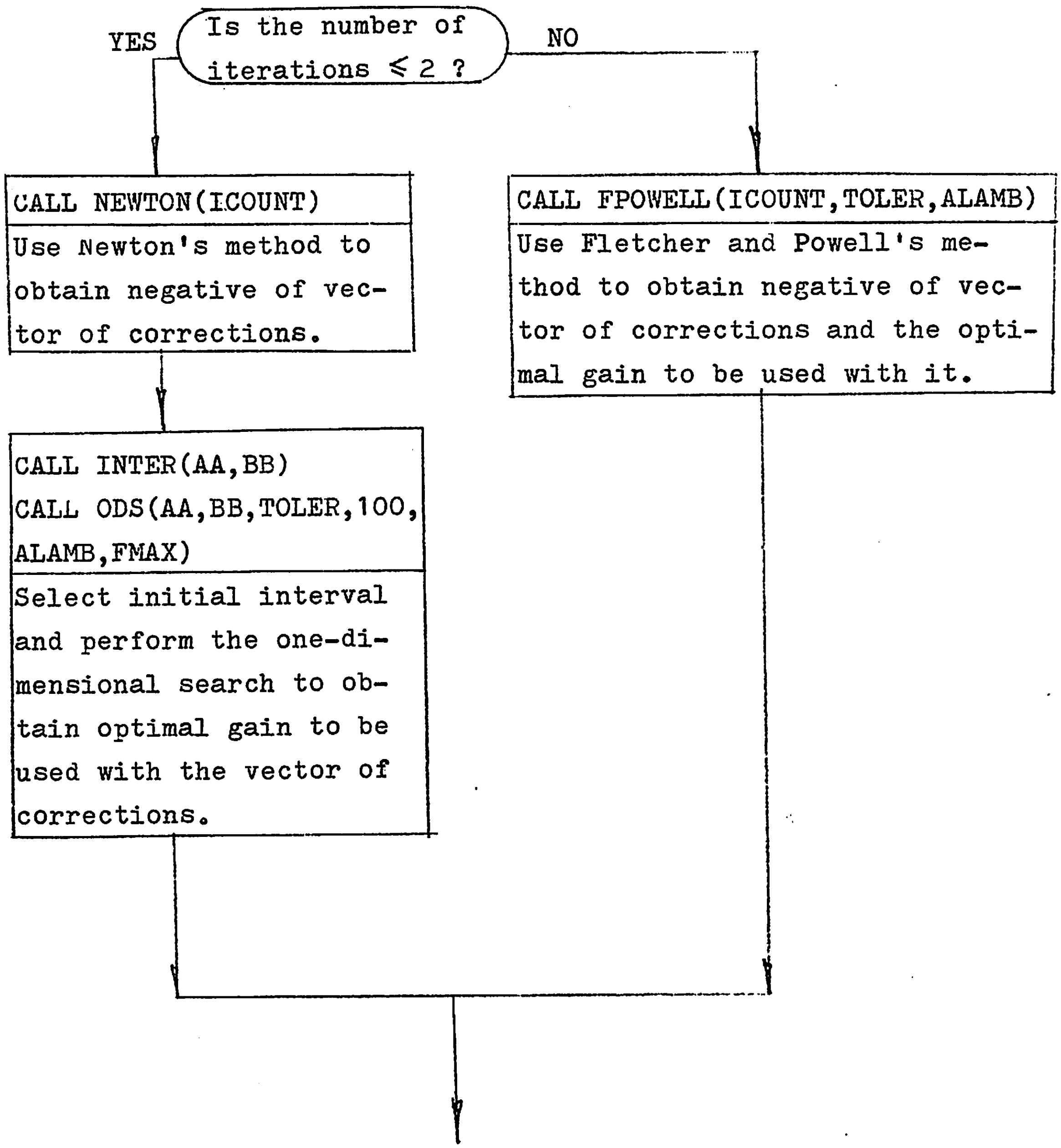
```
C     ********************************************************************
       CALL INTER(AA,BB)
       CALL ODS(AA,BB,TOLER,100,ALAMB,FMAX)
C     ********************************************************************
C     *OBTAIN NEGATIVE OF CORRECTIONS AND CORRECT VECTOR X.               *
C     ********************************************************************
       DO 20 J=1,N
       D=ALAMB*DELX(J)
       X(J)=X(J)-D
  20   CONTINUE
C     ********************************************************************
C     *COMPUTE NEW ERROR.                                                 *
C     ********************************************************************
       CALL REST(ENORMSQ)
       ENA=2.*MD
       EPU=SQRT(ENORMSQ/ENA)
C     ********************************************************************
C     *RESTITUTE ENTIRE DIMENSION IN X FOR PRINTING PURPOSES.             *
C     ********************************************************************
       N=NN
C     ********************************************************************
C     *CHECK IF THE DIFFERENCE BETWEEN NEW AND PAST ERROR IS LESS THAN    *
C     *OR EQUAL TO E-10 FOR TWO CONSECUTIVE TIMES.                        *
C     ********************************************************************
       DEPU=ABS(EPU-EPUS)
       IF(DEPU.LE.1.0E-10)11,13
  11   IE=IE+1
       IF(IE.GE.2)GO TO 10
       IF(IE.EQ.1)GO TO 12
  13   IE=0
  12   EPUS=EPU
C     ********************************************************************
C     *CHECK IF THE VALUE OF THE ERROR IS LESS THAN OR EQUAL TO THE DESI-*
C     *RED MINIMUM.                                                       *
C     ********************************************************************
       IF(ABS(EPU).LE.EPSI)GO TO 10
C     ********************************************************************
C     *CHECK NUMBER OF ITERATIONS.                                        *
C     ********************************************************************
       M=M-1
       IF(M.LE.0) GO TO 35
       CALL SECOND(T2)
C     ********************************************************************
C     *PRINT NEW RMS PER UNIT ERROR.                                      *
C     ********************************************************************
       PRINT 9,ICOUNT
       PRINT 5,EPU
       TIME=TIME+T2-T1
       PRINT 30,TIME
       PRINT 4008
 4008  FORMAT(//)
C     ********************************************************************
C     *SET DIMENSION OF X TO N-1 AND PERFORM ANOTHER ITERATION.           *
C     ********************************************************************
       N=N-1
```

```
      GO TO 7
   35 PRINT 6, ALAMB
    6 FORMAT(* MORE ITERATIONS NEEDED LAMBDA=*,F10.4)
   10 CONTINUE
      CALL SECOND(T2)
      TIME=TIME+T2-T1
      AVEANG=0.
C     ******************************************************************
C     *COMPUTE EXCITATION VOLTAGES AND EXCITATION ANGLES.              *
C     ******************************************************************
      DO 52 J=1,NM
      VMR=ZMR(J)*UMR(J)-ZMI(J)*UMI(J)+VREF(J)*COS(ANGTER(J))
      VMI=ZMI(J)*UMR(J)+ZMR(J)*UMI(J)+VREF(J)*SIN(ANGTER(J))
      VM(J)=SQRT(VMR**2+VMI**2)
      ANGMACH(J)=ATAN2(VMI,VMR)
C     ******************************************************************
C     *CALCULATE AVERAGE ANGLE.                                        *
C     ******************************************************************
52    AVEANG=AVEANG+ROIN(J)*ANGMACH(J)/SUMIN
C     ******************************************************************
C     *CORRECT TERMINAL AND EXCITATION ANGLES.                         *
C     ******************************************************************
      DO 59 J=1,MD
59    ANGTER(J)=ANGTER(J)-AVEANG+0.3
      DO 60 J=1,NM
60    ANGMACH(J)=ANGMACH(J)-AVEANG+0.3
      PRINT 3006
3006  FORMAT(53X,* OPERATING STATE*/)
      PRINT 53
53    FORMAT(* NODE   TERM VOLTS   TERM ANG   EXC VOLTS    EXC ANGLE
     1 POWERS         VARS           VAMPS          PTAR           QTAR           VATA
     1R*)
C     ******************************************************************
C     *PRINT OPERATING STATE.                                          *
C     ******************************************************************
      DO 54 J=1,MD
      PRINT 56,J,VREF(J),ANGTER(J),VM(J),ANGMACH(J),POWERS(J),VARS(J),VA
     1MPS(J),PTAR(J),QTAR(J),VATAR(J)
54    CONTINUE
      PRINT 3003
      PRINT 9,ICOUNT
      PRINT 2005,EPU
      PRINT 30,TIME
C     ******************************************************************
C     *CALCULATE LOSSES.                                               *
C     ******************************************************************
      DO 70 J=1,MD
      DO 70 K=1,MD
70    TOTALPO=TOTALPO+RNR(K,J)*VREF(J)**2
      SUMLOSS=TOTALPI-TOTALPO
      PRINT 57,AVEANG
57    FORMAT(* AVERAGE ROTOR ANGLE =*,F10.6)
      PRINT 66,TOTALPI
66    FORMAT(* SUM OF POWER INPUTS =*,F15.6)
      PRINT 67,TOTALPO
```

```
 67   FORMAT(* SUM OF POWER OUTPUTS =*,F15.6)
      PRINT 58,SUMLOSS
 58   FORMAT(* SUM OF POWER LOSSES =*,F15.6)
C    *******************************************************************
C    *CHECK IF THERE IS ANOTHER ERROR FUNCTION TO USE.                  *
C    *******************************************************************
      IF(INDEX.LT.NF) GO TO 25
 5    FORMAT(* EPU =*,F10.5)
 2005 FORMAT(* MIN EPU =*,F10.5)
 9    FORMAT(* ITERATION NUMBER*,I3)
 30   FORMAT(* TIME AT THIS ITERATION =*,F10.5)
 31   FORMAT(* TIME SPENT BEFORE ITERATING =*,F10.5)
      RETURN
      END
```

PROGRAM NNEWT4

BLOCK S



```
J7512CS,7,060,70000.7512 ZARATE NNEWT4
RUN,S.
LGO.
'
      PROGRAM NNEWT4(INPUT,OUTPUT)
C    ******************************************************************
C    *THIS PROGRAM OBTAINS THE OPERATING STATE OF A POWER SYSTEM FROM AN*
C    *INITIAL ESTIMATED STATE. THE PROGRAM USES NEWTONS METHOD FOR THE  *
C    *FIRST TWO ITERATIONS AND THEN FLETCHER AND POWELLS METHOD, UNTIL  *
C    *SATISFACTORY MINIMIZATION OF ERROR FUNCTION IS ACCOMPLISHED.      *
C    ******************************************************************
      COMMON INDEX
      COMMON/FLET/G(20),S(20),AC(20,20),BC(20,20),ICOUNTF
      COMMON/REF/VREF(8),ANGTER(8),VR(8),VI(8),UMR(8),UMI(8)
      COMMON/POWER/TOTALPO,TOTALPI,POWERS(8),VARS(8),VAMPS(8)
      COMMON/SHARE/X(100),DEL(100),A(100,100),N
      COMMON/CRST/DELX(100),DELXO(100)
      COMMON/ADMIT/RNR(8,8),RNI(8,8),MD
      COMMON/COSTS/CP2(8),CQ2(8)
      COMMON/TAR/PTAR(8),QTAR(8),VATAR(8)
      COMMON/DERIV/DPDV(8,8),DPDA(8,8),DQDV(8,8),DQDA(8,8)
      COMMON/ERROR/PE(8),QE(8)
      DIMENSION ROIN(8),ZMR(8),ZMI(8),PSPIN(8),VM(8),ANGMACH(8)
      DIMENSION CP(8),CQ(8)
      CALL SECOND(T1)
C    ******************************************************************
C    *READ DIMENSION OF VECTOR X (N=2*MD), MAXIMUM NUMBER OF ITERATIONS,*
C    *SCRATCH VARIABLE, NUMBER OF ERROR FUNCTIONS TO USE, NUMBER OF NO- *
C    *DES, TOLERANCE FOR MINIMUM VALUE OF ERROR FUNCTION, AND TOLERANCE *
C    *FOR ONE-DIMENSIONAL SEARCH.                                       *
C    ******************************************************************
      READ 3,N,M,K,NF,MD,EPSI,TOLER
 3    FORMAT(5I5,2F10.6)
      NN=N
      INDEX=0
   25 INDEX=INDEX+1
C    ******************************************************************
C    *READ INITIAL ESTIMATE OF NODE VOLTAGES AND ANGLES GIVEN AS A VEC- *
C    *TOR X. THE FIRST MD ELEMENTS OF THIS VECTOR ARE THE NODE VOLTAGES,*
C    *AND THE LAST MD ELEMENTS ARE ANGLES. DIMENSION OF X IS THEN N.    *
C    ******************************************************************
      READ 2,(X(I),I=1,N)
    2 FORMAT(8F10.6)
C    ******************************************************************
C    *READ ADMITTANCE MATRIX OF REGULATED VOLTAGE NODES.                *
C    ******************************************************************
      DO 2000 J=1,MD
      DO 2000 I=1,MD
      READ 2001,RNR(I,J),RNI(I,J)
 2001 FORMAT(2F10.6)
 2000 CONTINUE
C    ******************************************************************
C    *READ SENSITIVITY COEFFICIENTS, AND POWER TARGETS.                 *
C    ******************************************************************
      DO 2002 J=1,MD
```

```
      READ 2003,CP(J),CQ(J),PTAR(J),QTAR(J)
 2003 FORMAT(4F14.9)
      VATAR(J)=SQRT(PTAR(J)**2+QTAR(J)**2)
      CP2(J)=CP(J)**2
      CQ2(J)=CQ(J)**2
 2002 CONTINUE
      CALL SECOND(PT1)
      PRINT 1
 1    FORMAT(1H1)
      PRINT 160
160   FORMAT(//45H ADMITTANCE MATRIX RN REGULATED VOLTAGE NODES    //)
      DO 164 KA=1,MD,4
      IF(KA+3-MD)162,161,161
161   KB=MD
      GO TO 163
162   KB=KA+3
163   PRINT 313,(JCH,JCH=KA,KB)
313   FORMAT(/4I30/)
      DO 164 J=1,MD
164   PRINT 316,J,(RNR(J,K),RNI(J,K),K=KA,KB)
316   FORMAT(I5,4(F17.2,F13.2))
      PRINT 4008
      PRINT 336
 336  FORMAT(20X,* MACHINE CHARACTERISTICS*/)
      PRINT 337
 337  FORMAT( 66H NODE          INERTIA        RESISTANCE        REACTANCE   MA
     1XIMUM POWER      )
      JCOUNT=0
      SUMIN=0.
C     ******************************************************************
C     *READ MACHINE CHARACTERISTICS, INERTIA, REACTANCE, MAXIMUM POWER,  *
C     *AND NUMBER OF MACHINES.                                           *
C     ******************************************************************
 45   READ 338,J,ROIN(J),ZMR(J),ZMI(J),PSPIN(J),NM
 338  FORMAT(I3,2X,4F15.6,10X,I5)
      SUMIN=SUMIN+ROIN(J)
      JCOUNT=JCOUNT+1
      PRINT 338,J,ROIN(J),ZMR(J),ZMI(J),PSPIN(J)
      IF(JCOUNT.LT.NM)GO TO 45
      PRINT 3000
 3000 FORMAT(///,53X,* INITIAL STATE*/)
C     ******************************************************************
C     *OBTAIN INITIAL ERROR.                                             *
C     ******************************************************************
      CALL REST(ENORMSQ)
      ENA=2.*MD
      EPU=SQRT(ENORMSQ/ENA)
      PRINT 3001
 3001 FORMAT(* NODE    TERM VOLTS    TERM ANG        CP              CQ
     1 POWERS          VARS            VAMPS          PTAR           QTAR            VATA
     1R*)
C     ******************************************************************
C     *PRINT INITIAL STATE.                                              *
C     ******************************************************************
      DO 3002 J=1,MD
```

```
3002 PRINT 56,J,VREF(J),ANGTER(J),CP(J),CQ(J),POWERS(J),VARS(J),VAMPS(J
    1),PTAR(J),QTAR(J),VATAR(J)
56   FORMAT(I3,4X,F10.6,2X,F10.6,2X,F10.6,2X,F10.6,2X,F10.4,2X,F10.4,2X
    1,F10.4,2X,F10.4,2X,F10.4,2X,F10.4)
     PRINT 3003
3003 FORMAT(/)
C    ******************************************************************
C    *PRINT INITIAL ERROR IN RMS PER UNIT VALUE.                      *
C    ******************************************************************
     PRINT 5,EPU
     CALL SECOND(PT2)
     ICOUNT=0
     IE=0
     EPUS=0.
C    ******************************************************************
C    *TAKE ONLY (2*MD-1) ELEMENTS IN X BECAUSE OF REFERENCE ANGLE.    *
C    ******************************************************************
     N=N-1
     CALL SECOND(T2)
     TIME=T2-T1-PT2+PT1
     PRINT 31,TIME
     PRINT 4008
     ICOUNTF=0
 7   CALL SECOND(T1)
C    ******************************************************************
C    *OBTAIN THE GRADIENT OF ERROR FUNCTION.                          *
C    ******************************************************************
     CALL GRAD1
     ICOUNT=ICOUNT+1
     IF(ICOUNT.LE.2)100,101
C    ******************************************************************
C    *USE NEWTONS METHOD FOR FIRST TWO ITERATIONS.                    *
C    ******************************************************************
 100 CONTINUE
     CALL NEWTON(ICOUNT)
C    ******************************************************************
C    *PERFORM A ONE-DIMENSIONAL SEARCH TO OBTAIN OPTIMAL LAMBDA.      *
C    ******************************************************************
     CALL INTER(AA,BB)
     CALL ODS(AA,BB,TOLER,100,ALAMB,FMAX)
     GO TO 261
C    ******************************************************************
C    *USE FLETCHER AND POWELLS METHOD FOR THE REMAINING ITERATIONS.   *
C    ******************************************************************
 101 CONTINUE
     CALL FPOWELL(ICOUNT,TOLER,ALAMB)
 261 CONTINUE
     DO 20 J=1,N
C    ******************************************************************
C    *OBTAIN NEGATIVE OF CORRECTIONS AND CORRECT VECTOR X.            *
C    ******************************************************************
     D=ALAMB*DELX(J)
     X(J)=X(J)-D
 20  CONTINUE
C    ******************************************************************
```

```
C     *COMPUTE NEW ERROR.                                                    *
C     ************************************************************************
       CALL REST(ENORMSQ)
       ENA=2.*MD
       EPU=SQRT(ENORMSQ/ENA)
C     ************************************************************************
C     *RESTITUTE ENTIRE DIMENSION IN X FOR PRINTING PURPOSES.                *
C     ************************************************************************
       N=NN
C     ************************************************************************
C     *CHECK IF THE DIFFERENCE BETWEEN NEW AND PAST ERROR IS LESS THAN       *
C     *OR EQUAL TO E-10 FOR TWO CONSECUTIVE TIMES.                           *
C     ************************************************************************
       DEPU=ABS(EPU-EPUS)
       IF(DEPU.LE.1.0E-10)11,13
 11    IE=IE+1
       IF(IE.GE.2)GO TO 10
       IF(IE.EQ.1)GO TO 12
 13    IE=0
 12    EPUS=EPU
C     ************************************************************************
C     *CHECK IF THE VALUE OF THE ERROR IS LESS THAN OR EQUAL TO THE DESI-*
C     *RED MINIMUM.                                                          *
C     ************************************************************************
       IF(ABS(EPU).LE.EPSI)GO TO 10
C     ************************************************************************
C     *CHECK NUMBER OF ITERATIONS.                                           *
C     ************************************************************************
       M=M-1
       IF(M.LE.0) GO TO 35
       CALL SECOND(T2)
C     ************************************************************************
C     *PRINT NEW RMS PER UNIT ERROR.                                         *
C     ************************************************************************
       PRINT 9,ICOUNT
       PRINT 5,EPU
       TIME=TIME+T2-T1
       PRINT 30,TIME
       PRINT 4008
 4008  FORMAT(//)
C     ************************************************************************
C     *SET DIMENSION OF X TO N-1 AND PERFORM ANOTHER ITERATION.              *
C     ************************************************************************
       N=N-1
       GO TO 7
    35 PRINT 6, ALAMB
     6 FORMAT(* MORE ITERATIONS NEEDED LAMBDA=*,F10.4)
    10 CONTINUE
       CALL SECOND(T2)
       TIME=TIME+T2-T1
       AVEANG=0.
C     ************************************************************************
C     *COMPUTE EXCITATION VOLTAGES AND EXCITATION ANGLES.                    *
C     ************************************************************************
       DO 52 J=1,NM
```

```
      VMR=ZMR(J)*UMR(J)-ZMI(J)*UMI(J)+VREF(J)*COS(ANGTER(J))
      VMI=ZMI(J)*UMR(J)+ZMR(J)*UMI(J)+VREF(J)*SIN(ANGTER(J))
      VM(J)=SQRT(VMR**2+VMI**2)
      ANGMACH(J)=ATAN2(VMI,VMR)
C     *****************************************************************
C     *CALCULATE AVERAGE ANGLE.                                        *
C     *****************************************************************
 52   AVEANG=AVEANG+ROIN(J)*ANGMACH(J)/SUMIN
C     *****************************************************************
C     *CORRECT TERMINAL AND EXCITATION ANGLES.                         *
C     *****************************************************************
      DO 59 J=1,MD
 59   ANGTER(J)=ANGTER(J)-AVEANG+0.3
      DO 60 J=1,NM
 60   ANGMACH(J)=ANGMACH(J)-AVEANG+0.3
      PRINT 3006
 3006 FORMAT(53X,* OPERATING STATE*/)
      PRINT 53
 53   FORMAT(* NODE   TERM VOLTS   TERM ANG   EXC VOLTS    EXC ANGLE
     1 POWERS         VARS          VAMPS          PTAR          QTAR          VATA
     1R*)
C     *****************************************************************
C     *PRINT OPERATING STATE.                                          *
C     *****************************************************************
      DO 54 J=1,MD
      PRINT 56,J,VREF(J),ANGTER(J),VM(J),ANGMACH(J),POWERS(J),VARS(J),VA
     1MPS(J),PTAR(J),QTAR(J),VATAR(J)
 54   CONTINUE
      PRINT 3003
      PRINT 9,ICOUNT
      PRINT 2005,EPU
      PRINT 30,TIME
C     *****************************************************************
C     *CALCULATE LOSSES.                                               *
C     *****************************************************************
      DO 70 J=1,MD
      DO 70 K=1,MD
 70   TOTALPO=TOTALPO+RNR(K,J)*VREF(J)**2
      SUMLOSS=TOTALPI-TOTALPO
      PRINT 57,AVEANG
 57   FORMAT(* AVERAGE ROTOR ANGLE =*,F10.6)
      PRINT 66,TOTALPI
 66   FORMAT(* SUM OF POWER INPUTS =*,F15.6)
      PRINT 67,TOTALPO
 67   FORMAT(* SUM OF POWER OUTPUTS =*,F15.6)
      PRINT 58,SUMLOSS
 58   FORMAT(* SUM OF POWER LOSSES =*,F15.6)
C     *****************************************************************
C     *CHECK IF THERE IS ANOTHER ERROR FUNCTION TO USE.                *
C     *****************************************************************
      IF(INDEX.LT.NF) GO TO 25
 5    FORMAT(* EPU =*,F10.5)
 2005 FORMAT(* MIN EPU =*,F10.5)
 9    FORMAT(* ITERATION NUMBER*,I3)
 30   FORMAT(* TIME AT THIS ITERATION =*,F10.5)
```

```
31    FORMAT(* TIME SPENT BEFORE ITERATING =*,F10.5)
      RETURN
      END
```

PROGRAM NNEWT5
BLOCK S

CALL FPOWELL(ICOUNT,TOLER,ALAMB)
Use Fletcher and Powell's method to obtain negative of vector of corrections and the optimal gain to be used with it.

```
J7512CS,7,060,70000.7512 ZARATE NNEWT5
RUN,S.
LGO.
'
      PROGRAM NNEWT5(INPUT,OUTPUT)
C     ******************************************************************
C     *THIS PROGRAM OBTAINS THE OPERATING STATE OF A POWER SYSTEM FROM AN*
C     *INITIAL ESTIMATED STATE. THE PROGRAM USES FLETCHER AND POWELLS ME-*
C     *THOD ONLY, UNTIL SATISFACTORY MINIMIZATION OF ERROR FUNCTION IS   *
C     *ACCOMPLISHED.                                                     *
C     ******************************************************************
      COMMON INDEX
      COMMON/FLET/G(20),S(20),AC(20,20),BC(20,20),ICOUNTF
      COMMON/SHARE/X(100),DEL(100),A(100,100),N
      COMMON/CRST/DELX(100),DELXO(100)
      COMMON/ADMIT/RNR(8,8),RNI(8,8),MD
      COMMON/COSTS/CP2(8),CQ2(8)
      COMMON/TAR/PTAR(8),QTAR(8),VATAR(8)
      COMMON/DERIV/DPDV(8,8),DPDA(8,8),DQDV(8,8),DQDA(8,8)
      COMMON/ERROR/PE(8),QE(8)
      COMMON/REF/VREF(8),ANGTER(8),VR(8),VI(8),UMR(8),UMI(8)
      COMMON/POWER/TOTALPO,TOTALPI,POWERS(8),VARS(8),VAMPS(8)
      DIMENSION ROIN(8),ZMR(8),ZMI(8),PSPIN(8),VM(8),ANGMACH(8)
      DIMENSION CP(8),CQ(8)
      CALL SECOND(T1)
C     ******************************************************************
C     *READ DIMENSION OF VECTOR X (N=2*MD), MAXIMUM NUMBER OF ITERATIONS,*
C     *SCRATCH VARIABLE, NUMBER OF ERROR FUNCTIONS TO USE, NUMBER OF NO- *
C     *DES, TOLERANCE FOR MINIMUM VALUE OF ERROR FUNCTION, AND TOLERANCE *
C     *FOR ONE-DIMENSIONAL SEARCH.                                       *
C     ******************************************************************
      READ 3,N,M,K,NF,MD,EPSI,TOLER
 3    FORMAT(5I5,2F10.6)
      NN=N
      INDEX=0
   25 INDEX=INDEX+1
C     ******************************************************************
C     *READ INITIAL ESTIMATE OF NODE VOLTAGES AND ANGLES GIVEN AS A VEC- *
C     *TOR X. THE FIRST MD ELEMENTS OF THIS VECTOR ARE THE NODE VOLTAGES,*
C     *AND THE LAST MD ELEMENTS ARE ANGLES. DIMENSION OF X IS THEN N.    *
C     ******************************************************************
      READ 2,(X(I),I=1,N)
    2 FORMAT(8F10.6)
C     ******************************************************************
C     *READ ADMITTANCE MATRIX OF REGULATED VOLTAGE NODES.                *
C     ******************************************************************
      DO 2000 J=1,MD
      DO 2000 I=1,MD
      READ 2001,RNR(I,J),RNI(I,J)
 2001 FORMAT(2F10.6)
 2000 CONTINUE
C     ******************************************************************
C     *READ SENSITIVITY COEFFICIENTS, AND POWER TARGETS.                 *
C     ******************************************************************
      DO 2002 J=1,MD
```

```
      READ 2003,CP(J),CQ(J),PTAR(J),QTAR(J)
 2003 FORMAT(4F14.9)
      VATAR(J)=SQRT(PTAR(J)**2+QTAR(J)**2)
      CP2(J)=CP(J)**2
      CQ2(J)=CQ(J)**2
 2002 CONTINUE
      CALL SECOND(PT1)
      PRINT 1
 1    FORMAT(1H1)
      PRINT 160
160   FORMAT(//45H ADMITTANCE MATRIX RN REGULATED VOLTAGE NODES   //)
      DO 164 KA=1,MD,4
      IF(KA+3-MD)162,161,161
161   KB=MD
      GO TO 163
162   KB=KA+3
163   PRINT 313,(JCH,JCH=KA,KB)
313   FORMAT(/4I30/)
      DO 164 J=1,MD
164   PRINT 316,J,(RNR(J,K),RNI(J,K),K=KA,KB)
316   FORMAT(I5,4(F17.2,F13.2))
      PRINT 4008
      PRINT 336
 336  FORMAT(20X,* MACHINE CHARACTERISTICS*/)
      PRINT 337
 337  FORMAT( 66H NODE        INERTIA      RESISTANCE      REACTANCE  MA
     1XIMUM POWER     )
      JCOUNT=0
      SUMIN=0.
C     ******************************************************************
C     *READ MACHINE CHARACTERISTICS, INERTIA, REACTANCE, MAXIMUM POWER, *
C     *AND NUMBER OF MACHINES.                                          *
C     ******************************************************************
 45   READ 338,J,ROIN(J),ZMR(J),ZMI(J),PSPIN(J),NM
 338  FORMAT(I3,2X,4F15.6,10X,I5)
      SUMIN=SUMIN+ROIN(J)
      JCOUNT=JCOUNT+1
      PRINT 338,J,ROIN(J),ZMR(J),ZMI(J),PSPIN(J)
      IF(JCOUNT.LT.NM)GO TO 45
      PRINT 3000
 3000 FORMAT(///,53X,* INITIAL STATE*/)
C     ******************************************************************
C     *OBTAIN INITIAL ERROR.                                            *
C     ******************************************************************
      CALL REST(ENORMSQ)
      ENA=2.*MD
      EPU=SQRT(ENORMSQ/ENA)
      PRINT 3001
 3001 FORMAT(* NODE   TERM VOLTS   TERM ANG       CP             CQ
     1 POWERS        VARS           VAMPS          PTAR           QTAR          VATA
     1R*)
C     ******************************************************************
C     *PRINT INITIAL STATE.                                             *
C     ******************************************************************
      DO 3002 J=1,MD
```

```
3002 PRINT 56,J,VREF(J),ANGTER(J),CP(J),CQ(J),POWERS(J),VARS(J),VAMPS(J
    1),PTAR(J),QTAR(J),VATAR(J)
 56  FORMAT(I3,4X,F10.6,2X,F10.6,2X,F10.6,2X,F10.6,2X,F10.4,2X,F10.4,2X
    1,F10.4,2X,F10.4,2X,F10.4,2X,F10.4)
     PRINT 3003
3003 FORMAT(/)
C    *****************************************************************
C    *PRINT INITIAL ERROR IN RMS PER UNIT VALUE.                     *
C    *****************************************************************
     PRINT 5,EPU
     CALL SECOND(PT2)
     ICOUNT=0
     IE=0
     EPUS=0.
C    *****************************************************************
C    *TAKE ONLY (2*MD-1) ELEMENTS IN X BECAUSE OF REFERENCE ANGLE.   *
C    *****************************************************************
     N=N-1
     CALL SECOND(T2)
     TIME=T2-T1-PT2+PT1
     PRINT 31,TIME
     PRINT 4008
     ICOUNTF=0
 7   CALL SECOND(T1)
C    *****************************************************************
C    *OBTAIN THE GRADIENT OF ERROR FUNCTION.                         *
C    *****************************************************************
     CALL GRAD1
     ICOUNT=ICOUNT+1
C    *****************************************************************
C    *USE FLETCHER AND POWELLS METHOD ONLY.                          *
C    *****************************************************************
     CALL FPOWELL(ICOUNT,TOLER,ALAMB)
C    *****************************************************************
C    *OBTAIN NEGATIVE OF CORRECTIONS AND CORRECT VECTOR X.           *
C    *****************************************************************
     DO 20 J=1,N
     D=ALAMB*DELX(J)
     X(J)=X(J)-D
 20  CONTINUE
C    *****************************************************************
C    *COMPUTE NEW ERROR.                                             *
C    *****************************************************************
     CALL REST(ENORMSQ)
     ENA=2.*MD
     EPU=SQRT(ENORMSQ/ENA)
C    *****************************************************************
C    *RESTITUTE ENTIRE DIMENSION IN X FOR PRINTING PURPOSES.         *
C    *****************************************************************
     N=NN
C    *****************************************************************
C    *CHECK IF THE DIFFERENCE BETWEEN NEW AND PAST ERROR IS LESS THAN *
C    *OR EQUAL TO E-10 FOR TWO CONSECUTIVE TIMES.                    *
C    *****************************************************************
     DEPU=ABS(EPU-EPUS)
```

```
      IF(DEPU.LE.1.0E-10)11,13
 11   IE=IE+1
      IF(IE.GE.2)GO TO 10
      IF(IE.EQ.1)GO TO 12
 13   IE=0
 12   EPUS=EPU
C     ******************************************************************
C     *CHECK IF THE VALUE OF THE ERROR IS LESS THAN OR EQUAL TO THE DESI-*
C     *RED MINIMUM.                                                     *
C     ******************************************************************
      IF(ABS(EPU).LE.EPSI)GO TO 10
C     ******************************************************************
C     *CHECK NUMBER OF ITERATIONS.                                      *
C     ******************************************************************
      M=M-1
      IF(M.LE.0) GO TO 35
      CALL SECOND(T2)
C     ******************************************************************
C     *PRINT NEW RMS PER UNIT ERROR.                                    *
C     ******************************************************************
      PRINT 9,ICOUNT
      PRINT 5,EPU
      TIME=TIME+T2-T1
      PRINT 30,TIME
      PRINT 4008
 4008 FORMAT(//)
C     ******************************************************************
C     *SET DIMENSION OF X TO N-1 AND PERFORM ANOTHER ITERATION.         *
C     ******************************************************************
      N=N-1
      GO TO 7
   35 PRINT 6, ALAMB
    6 FORMAT(* MORE ITERATIONS NEEDED LAMBDA=*,F10.4)
   10 CONTINUE
      CALL SECOND(T2)
      TIME=TIME+T2-T1
      AVEANG=0.
C     ******************************************************************
C     *COMPUTE EXCITATION VOLTAGES AND EXCITATION ANGLES.               *
C     ******************************************************************
      DO 52 J=1,NM
      VMR=ZMR(J)*UMR(J)-ZMI(J)*UMI(J)+VREF(J)*COS(ANGTER(J))
      VMI=ZMI(J)*UMR(J)+ZMR(J)*UMI(J)+VREF(J)*SIN(ANGTER(J))
      VM(J)=SQRT(VMR**2+VMI**2)
      ANGMACH(J)=ATAN2(VMI,VMR)
C     ******************************************************************
C     *CALCULATE AVERAGE ANGLE.                                         *
C     ******************************************************************
 52   AVEANG=AVEANG+ROIN(J)*ANGMACH(J)/SUMIN
C     ******************************************************************
C     *CORRECT TERMINAL AND EXCITATION ANGLES.                          *
C     ******************************************************************
      DO 59 J=1,MD
 59   ANGTER(J)=ANGTER(J)-AVEANG+0.3
      DO 60 J=1,NM
```

```
60    ANGMACH(J)=ANGMACH(J)-AVEANG+0.3
      PRINT 3006
3006 FORMAT(53X,* OPERATING STATE*/)
      PRINT 53
53    FORMAT(* NODE    TERM VOLTS    TERM ANG    EXC VOLTS     EXC ANGLE
     1 POWERS         VARS            VAMPS           PTAR            QTAR            VATA
     1R*)
C     ******************************************************************
C     *PRINT OPERATING STATE.                                          *
C     ******************************************************************
      DO 54 J=1,MD
      PRINT 56,J,VREF(J),ANGTER(J),VM(J),ANGMACH(J),POWERS(J),VARS(J),VA
     1MPS(J),PTAR(J),QTAR(J),VATAR(J)
54    CONTINUE
      PRINT 3003
      PRINT 9,ICOUNT
      PRINT 2005,EPU
      PRINT 30,TIME
C     ******************************************************************
C     *CALCULATE LOSSES.                                               *
C     ******************************************************************
      DO 70 J=1,MD
      DO 70 K=1,MD
70    TOTALPO=TOTALPO+RNR(K,J)*VREF(J)**2
      SUMLOSS=TOTALPI-TOTALPO
      PRINT 57,AVEANG
57    FORMAT(* AVERAGE ROTOR ANGLE =*,F10.6)
      PRINT 66,TOTALPI
66    FORMAT(* SUM OF POWER INPUTS =*,F15.6)
      PRINT 67,TOTALPO
67    FORMAT(* SUM OF POWER OUTPUTS =*,F15.6)
      PRINT 58,SUMLOSS
58    FORMAT(* SUM OF POWER LOSSES =*,F15.6)
C     ******************************************************************
C     *CHECK IF THERE IS ANOTHER ERROR FUNCTION TO USE.                *
C     ******************************************************************
      IF(INDEX.LT.NF) GO TO 25
5     FORMAT(* EPU =*,F10.5)
2005 FORMAT(* MIN EPU =*,F10.5)
9     FORMAT(* ITERATION NUMBER*,I3)
30    FORMAT(* TIME AT THIS ITERATION =*,F10.5)
31    FORMAT(* TIME SPENT BEFORE ITERATING =*,F10.5)
      RETURN
      END
```

PROGRAM NNEWT6

BLOCK S

CALL INTER(AA,BB) CALL ODS(AA,BB,TOLER,100, ALAMB,FMAX)
Select initial interval and perform the one-dimensional search to obtain the optimal gain to be used with the vector of corrections.

```
J7512CS,7,060,70000.7512 ZARATE NNEWT6
RUN,S.
LGO.
'
      PROGRAM NNEWT6(INPUT,OUTPUT)
C    ******************************************************************
C    *THIS PROGRAM OBTAINS THE OPERATING STATE OF A POWER SYSTEM FROM AN*
C    *INITIAL ESTIMATED STATE. THE PROGRAM USES STEEPEST DESCENT ONLY,  *
C    *UNTIL SATISFACTORY MINIMIZATION OF ERROR FUNCTION IS ACCOMPLISHED.*
C    ******************************************************************
      COMMON INDEX
      COMMON/SHARE/X(100),DEL(100),A(100,100),N
      COMMON/CRST/DELX(100),DELX0(100)
      COMMON/ADMIT/RNR(8,8),RNI(8,8),MD
      COMMON/COSTS/CP2(8),CQ2(8)
      COMMON/TAR/PTAR(8),QTAR(8),VATAR(8)
      COMMON/DERIV/DPDV(8,8),DPDA(8,8),DQDV(8,8),DQDA(8,8)
      COMMON/ERROR/PE(8),QE(8)
      COMMON/REF/VREF(8),ANGTER(8),VR(8),VI(8),UMR(8),UMI(8)
      COMMON/POWER/TOTALPO,TOTALPI,POWERS(8),VARS(8),VAMPS(8)
      DIMENSION ROIN(8),ZMR(8),ZMI(8),PSPIN(8),VM(8),ANGMACH(8)
      DIMENSION CP(8),CQ(8)
      CALL SECOND(T1)
C    ******************************************************************
C    *READ DIMENSION OF VECTOR X (N=2*MD), MAXIMUM NUMBER OF ITERATIONS,*
C    *SCRATCH VARIABLE, NUMBER OF ERROR FUNCTIONS TO USE, NUMBER OF NO- *
C    *DES, TOLERANCE FOR MINIMUM VALUE OF ERROR FUNCTION, AND TOLERANCE *
C    *FOR ONE-DIMENSIONAL SEARCH.                                       *
C    ******************************************************************
      READ 3,N,M,K,NF,MD,EPSI,TOLER
 3    FORMAT(5I5,2F10.6)
      NN=N
      INDEX=0
   25 INDEX=INDEX+1
C    ******************************************************************
C    *READ INITIAL ESTIMATE OF NODE VOLTAGES AND ANGLES GIVEN AS A VEC- *
C    *TOR X. THE FIRST MD ELEMENTS OF THIS VECTOR ARE THE NODE VOLTAGES,*
C    *AND THE LAST MD ELEMENTS ARE ANGLES. DIMENSION OF X IS THEN N.    *
C    ******************************************************************
      READ 2,(X(I),I=1,N)
    2 FORMAT(8F10.6)
C    ******************************************************************
C    *READ ADMITTANCE MATRIX OF REGULATED VOLTAGE NODES.                *
C    ******************************************************************
      DO 2000 J=1,MD
      DO 2000 I=1,MD
      READ 2001,RNR(I,J),RNI(I,J)
 2001 FORMAT(2F10.6)
 2000 CONTINUE
C    ******************************************************************
C    *READ SENSITIVITY COEFFICIENTS, AND POWER TARGETS.                 *
C    ******************************************************************
      DO 2002 J=1,MD
      READ 2003,CP(J),CQ(J),PTAR(J),QTAR(J)
 2003 FORMAT(4F14.9)
```

```
      VATAR(J)=SQRT(PTAR(J)**2+QTAR(J)**2)
      CP2(J)=CP(J)**2
      CQ2(J)=CQ(J)**2
 2002 CONTINUE
      CALL SECOND(PT1)
      PRINT 1
 1    FORMAT(1H1)
      PRINT 160
160   FORMAT(//45H ADMITTANCE MATRIX RN REGULATED VOLTAGE NODES   //)
      DO 164 KA=1,MD,4
      IF(KA+3-MD)162,161,161
161   KB=MD
      GO TO 163
162   KB=KA+3
163   PRINT 313,(JCH,JCH=KA,KB)
313   FORMAT(/4I30/)
      DO 164 J=1,MD
164   PRINT 316,J,(RNR(J,K),RNI(J,K),K=KA,KB)
316   FORMAT(I5,4(F17.2,F13.2))
      PRINT 4008
      PRINT 336
 336  FORMAT(20X,* MACHINE CHARACTERISTICS*/)
      PRINT 337
 337  FORMAT( 66H NODE          INERTIA        RESISTANCE         REACTANCE  MA
     1XIMUM POWER     )
      JCOUNT=0
      SUMIN=0.
C     ******************************************************************
C     *READ MACHINE CHARACTERISTICS, INERTIA, REACTANCE, MAXIMUM POWER,  *
C     *AND NUMBER OF MACHINES.                                           *
C     ******************************************************************
 45   READ 338,J,ROIN(J),ZMR(J),ZMI(J),PSPIN(J),NM
 338  FORMAT(I3,2X,4F15.6,10X,I5)
      SUMIN=SUMIN+ROIN(J)
      JCOUNT=JCOUNT+1
      PRINT 338,J,ROIN(J),ZMR(J),ZMI(J),PSPIN(J)
      IF(JCOUNT.LT.NM)GO TO 45
      PRINT 3000
 3000 FORMAT(///,53X,* INITIAL STATE*/)
C     ******************************************************************
C     *OBTAIN INITIAL ERROR.                                             *
C     ******************************************************************
      CALL REST(ENORMSQ)
      ENA=2.*MD
      EPU=SQRT(ENORMSQ/ENA)
      PRINT 3001
 3001 FORMAT(* NODE   TERM VOLTS   TERM ANG       CP              CQ
     1 POWERS        VARS          VAMPS          PTAR          QTAR          VATA
     1R*)
C     ******************************************************************
C     *PRINT INITIAL STATE.                                              *
C     ******************************************************************
      DO 3002 J=1,MD
 3002 PRINT 56,J,VREF(J),ANGTER(J),CP(J),CQ(J),POWERS(J),VARS(J),VAMPS(J
     1),PTAR(J),QTAR(J),VATAR(J)
```

```
 56    FORMAT(I3,4X,F10.6,2X,F10.6,2X,F10.6,2X,F10.6,2X,F10.4,2X,F10.4,2X
      1,F10.4,2X,F10.4,2X,F10.4,2X,F10.4)
       PRINT 3003
 3003 FORMAT(/)
C     ******************************************************************
C     *PRINT INITIAL ERROR IN RMS PER UNIT VALUE.                      *
C     ******************************************************************
       PRINT 5,EPU
       CALL SECOND(PT2)
       ICOUNT=0
       IE=0
       EPUS=0.
C     ******************************************************************
C     *TAKE ONLY (2*MD-1) ELEMENTS IN X BECAUSE OF REFERENCE ANGLE.    *
C     ******************************************************************
       N=N-1
       CALL SECOND(T2)
       TIME=T2-T1-PT2+PT1
       PRINT 31,TIME
       PRINT 4008
 7     CALL SECOND(T1)
C     ******************************************************************
C     *USE STEEPEST DESCENT ONLY.                                      *
C     *OBTAIN THE GRADIENT OF ERROR FUNCTION.                          *
C     ******************************************************************
       CALL GRAD1
       ICOUNT=ICOUNT+1
C     ******************************************************************
C     *PERFORM A ONE-DIMENSIONAL SEARCH TO OBTAIN OPTIMAL LAMBDA.      *
C     ******************************************************************
       CALL INTER(AA,BB)
       CALL ODS(AA,BB,TOLER,100,ALAMB,FMAX)
C     ******************************************************************
C     *OBTAIN NEGATIVE OF CORRECTIONS AND CORRECT VECTOR X.            *
C     ******************************************************************
       DO 20 J=1,N
       D=ALAMB*DELX(J)
       X(J)=X(J)-D
 20    CONTINUE
C     ******************************************************************
C     *COMPUTE NEW ERROR.                                              *
C     ******************************************************************
       CALL REST(ENORMSQ)
       ENA=2.*MD
       EPU=SQRT(ENORMSQ/ENA)
C     ******************************************************************
C     *RESTITUTE ENTIRE DIMENSION IN X FOR PRINTING PURPOSES.          *
C     ******************************************************************
       N=NN
C     ******************************************************************
C     *CHECK IF THE DIFFERENCE BETWEEN NEW AND PAST ERROR IS LESS THAN *
C     *OR EQUAL TO E-10 FOR TWO CONSECUTIVE TIMES.                     *
C     ******************************************************************
       DEPU=ABS(EPU-EPUS)
       IF(DEPU.LE.1.0E-10)11,13
```

```
 11    IE=IE+1
       IF(IE.GE.2)GO TO 10
       IF(IE.EQ.1)GO TO 12
 13    IE=0
 12    EPUS=EPU
C     *****************************************************************
C     *CHECK IF THE VALUE OF THE ERROR IS LESS THAN OR EQUAL TO THE DESI-*
C     *RED MINIMUM.                                                     *
C     *****************************************************************
       IF(ABS(EPU).LE.EPSI)GO TO 10
C     *****************************************************************
C     *CHECK NUMBER OF ITERATIONS.                                      *
C     *****************************************************************
       M=M-1
       IF(M.LE.0) GO TO 35
       CALL SECOND(T2)
C     *****************************************************************
C     *PRINT NEW RMS PER UNIT ERROR.                                    *
C     *****************************************************************
       PRINT 9,ICOUNT
       PRINT 5,EPU
       TIME=TIME+T2-T1
       PRINT 30,TIME
       PRINT 4008
 4008  FORMAT(//)
C     *****************************************************************
C     *SET DIMENSION OF X TO N-1 AND PERFORM ANOTHER ITERATION.         *
C     *****************************************************************
       N=N-1
       GO TO 7
    35 PRINT 6, ALAMB
     6 FORMAT(* MORE ITERATIONS NEEDED LAMBDA=*,F10.4)
    10 CONTINUE
       CALL SECOND(T2)
       TIME=TIME+T2-T1
       AVEANG=0.
C     *****************************************************************
C     *COMPUTE EXCITATION VOLTAGES AND EXCITATION ANGLES.               *
C     *****************************************************************
       DO 52 J=1,NM
       VMR=ZMR(J)*UMR(J)-ZMI(J)*UMI(J)+VREF(J)*COS(ANGTER(J))
       VMI=ZMI(J)*UMR(J)+ZMR(J)*UMI(J)+VREF(J)*SIN(ANGTER(J))
       VM(J)=SQRT(VMR**2+VMI**2)
       ANGMACH(J)=ATAN2(VMI,VMR)
C     *****************************************************************
C     *CALCULATE AVERAGE ANGLE.                                         *
C     *****************************************************************
 52    AVEANG=AVEANG+ROIN(J)*ANGMACH(J)/SUMIN
C     *****************************************************************
C     *CORRECT TERMINAL AND EXCITATION ANGLES.                          *
C     *****************************************************************
       DO 59 J=1,MD
 59    ANGTER(J)=ANGTER(J)-AVEANG+0.3
       DO 60 J=1,NM
 60    ANGMACH(J)=ANGMACH(J)-AVEANG+0.3
```

```
      PRINT 3006
 3006 FORMAT(53X,* OPERATING STATE*/)
      PRINT 53
 53   FORMAT(* NODE   TERM VOLTS   TERM ANG   EXC VOLTS    EXC ANGLE
     1 POWERS       VARS          VAMPS          PTAR          QTAR          VATA
     1R*)
C     ************************************************************************
C     *PRINT OPERATING STATE.                                                *
C     ************************************************************************
      DO 54 J=1,MD
      PRINT 56,J,VREF(J),ANGTER(J),VM(J),ANGMACH(J),POWERS(J),VARS(J),VA
     1MPS(J),PTAR(J),QTAR(J),VATAR(J)
 54   CONTINUE
      PRINT 3003
      PRINT 9,ICOUNT
      PRINT 2005,EPU
      PRINT 30,TIME
C     ************************************************************************
C     *CALCULATE LOSSES.                                                     *
C     ************************************************************************
      DO 70 J=1,MD
      DO 70 K=1,MD
 70   TOTALPO=TOTALPO+RNR(K,J)*VREF(J)**2
      SUMLOSS=TOTALPI-TOTALPO
      PRINT 57,AVEANG
 57   FORMAT(* AVERAGE ROTOR ANGLE =*,F10.6)
      PRINT 66,TOTALPI
 66   FORMAT(* SUM OF POWER INPUTS =*,F15.6)
      PRINT 67,TOTALPO
 67   FORMAT(* SUM OF POWER OUTPUTS =*,F15.6)
      PRINT 58,SUMLOSS
 58   FORMAT(* SUM OF POWER LOSSES =*,F15.6)
C     ************************************************************************
C     *CHECK IF THERE IS ANOTHER ERROR FUNCTION TO USE.                      *
C     ************************************************************************
      IF(INDEX.LT.NF) GO TO 25
 5    FORMAT(* EPU =*,F10.5)
 2005 FORMAT(* MIN EPU =*,F10.5)
 9    FORMAT(* ITERATION NUMBER*,I3)
 30   FORMAT(* TIME AT THIS ITERATION =*,F10.5)
 31   FORMAT(* TIME SPENT BEFORE ITERATING =*,F10.5)
      RETURN
      END
```

PROGRAM NNEWT7

BLOCK S

CALL ESPD(K)

Compute matrix of second partial derivatives of error function with respect to adjustables.

CALL MTINV1(BB,O,D,IRROR,IPIV,IND)

Invert matrix of second partial derivatives.

CALL METHOD(IRROR,ICOUNT)

Check if it is possible to use the second order move vector. If not use Newton's method to obtain negative of corrections.

CALL INTER(AA,BB)
CALL ODS(AA,BB,TOLER,100,ALAMB,FMAX)

Select initial interval and perform the one-dimensional search to obtain optimal gain to be used with vector of corrections.

```
J7512CS,7,060,70000.7512 ZARATE NNEWT7
RUN,S.
LGO.
'
      PROGRAM NNEWT7(INPUT,OUTPUT)
C     ****************************************************************
C     *THIS PROGRAM OBTAINS THE OPERATING STATE OF A POWER SYSTEM FROM AN*
C     *INITIAL ESTIMATED STATE. THE PROGRAM USES THE SECOND ORDER MOVE  *
C     *VECTOR METHOD ONLY, UNTIL SATISFACTORY MINIMIZATION OF ERROR FUNC-*
C     *TION IS ACCOMPLISHED.                                            *
C     ****************************************************************
      COMMON INDEX
      COMMON/SHARE/X(100),DEL(100),A(100,100),N
      COMMON/CRST/DELX(100),DELXO(100)
      COMMON/ADMIT/RNR(8,8),RNI(8,8),MD
      COMMON/COSTS/CP2(8),CQ2(8)
      COMMON/TAR/PTAR(8),QTAR(8),VATAR(8)
      COMMON/DERIV/DPDV(8,8),DPDA(8,8),DQDV(8,8),DQDA(8,8)
      COMMON/ERROR/PE(8),QE(8)
      DIMENSION B(100)
      DIMENSION BB(16,1),IPIV(16),IND(16,2)
      COMMON/REF/VREF(8),ANGTER(8),VR(8),VI(8),UMR(8),UMI(8)
      COMMON/POWER/TOTALPO,TOTALPI,POWERS(8),VARS(8),VAMPS(8)
      DIMENSION ROIN(8),ZMR(8),ZMI(8),PSPIN(8),VM(8),ANGMACH(8)
      DIMENSION CP(8),CQ(8)
      CALL SECOND(T1)
C     ****************************************************************
C     *READ DIMENSION OF VECTOR X (N=2*MD), MAXIMUM NUMBER OF ITERATIONS,*
C     *SCRATCH VARIABLE, NUMBER OF ERROR FUNCTIONS TO USE, NUMBER OF NO- *
C     *DES, TOLERANCE FOR MINIMUM VALUE OF ERROR FUNCTION, AND TOLERANCE *
C     *FOR ONE-DIMENSIONAL SEARCH.                                      *
C     ****************************************************************
      READ 3,N,M,K,NF,MD,EPSI,TOLER
 3    FORMAT(5I5,2F10.6)
      NN=N
      INDEX=0
   25 INDEX=INDEX+1
C     ****************************************************************
C     *READ INITIAL ESTIMATE OF NODE VOLTAGES AND ANGLES GIVEN AS A VEC- *
C     *TOR X. THE FIRST MD ELEMENTS OF THIS VECTOR ARE THE NODE VOLTAGES,*
C     *AND THE LAST MD ELEMENTS ARE ANGLES. DIMENSION OF X IS THEN N.   *
C     ****************************************************************
      READ 2,(X(I),I=1,N)
    2 FORMAT(8F10.6)
C     ****************************************************************
C     *READ ADMITTANCE MATRIX OF REGULATED VOLTAGE NODES.               *
C     ****************************************************************
      DO 2000 J=1,MD
      DO 2000 I=1,MD
      READ 2001,RNR(I,J),RNI(I,J)
 2001 FORMAT(2F10.6)
 2000 CONTINUE
C     ****************************************************************
C     *READ SENSITIVITY COEFFICIENTS, AND POWER TARGETS.                *
C     ****************************************************************
```

```
      DO 2002 J=1,MD
      READ 2003,CP(J),CQ(J),PTAR(J),QTAR(J)
 2003 FORMAT(4F14.9)
      VATAR(J)=SQRT(PTAR(J)**2+QTAR(J)**2)
      CP2(J)=CP(J)**2
      CQ2(J)=CQ(J)**2
 2002 CONTINUE
      CALL SECOND(PT1)
      PRINT 1
 1    FORMAT(1H1)
      PRINT 160
160   FORMAT(//45H ADMITTANCE MATRIX RN REGULATED VOLTAGE NODES    //)
      DO 164 KA=1,MD,4
      IF(KA+3-MD)162,161,161
161   KB=MD
      GO TO 163
162   KB=KA+3
163   PRINT 313,(JCH,JCH=KA,KB)
313   FORMAT(/4I30/)
      DO 164 J=1,MD
164   PRINT 316,J,(RNR(J,K),RNI(J,K),K=KA,KB)
316   FORMAT(I5,4(F17.2,F13.2))
      PRINT 4008
      PRINT 336
 336  FORMAT(20X,* MACHINE CHARACTERISTICS*/)
      PRINT 337
 337  FORMAT( 66H NODE         INERTIA       RESISTANCE        REACTANCE  MA
     1XIMUM POWER     )
      JCOUNT=0
      SUMIN=0.
C     ******************************************************************
C     *READ MACHINE CHARACTERISTICS, INERTIA, REACTANCE, MAXIMUM POWER,  *
C     *AND NUMBER OF MACHINES.                                           *
C     ******************************************************************
 45   READ 338,J,ROIN(J),ZMR(J),ZMI(J),PSPIN(J),NM
 338  FORMAT(I3,2X,4F15.6,10X,I5)
      SUMIN=SUMIN+ROIN(J)
      JCOUNT=JCOUNT+1
      PRINT 338,J,ROIN(J),ZMR(J),ZMI(J),PSPIN(J)
      IF(JCOUNT.LT.NM)GO TO 45
      PRINT 3000
 3000 FORMAT(///,53X,* INITIAL STATE*/)
C     ******************************************************************
C     *OBTAIN INITIAL ERROR.                                             *
C     ******************************************************************
      CALL REST(ENORMSQ)
      ENA=2.*MD
      EPU=SQRT(ENORMSQ/ENA)
      PRINT 3001
 3001 FORMAT(* NODE   TERM VOLTS   TERM ANG       CP              CQ
     1 POWERS         VARS          VAMPS         PTAR          QTAR          VATA
     1R*)
C     ******************************************************************
C     *PRINT INITIAL STATE.                                              *
C     ******************************************************************
```

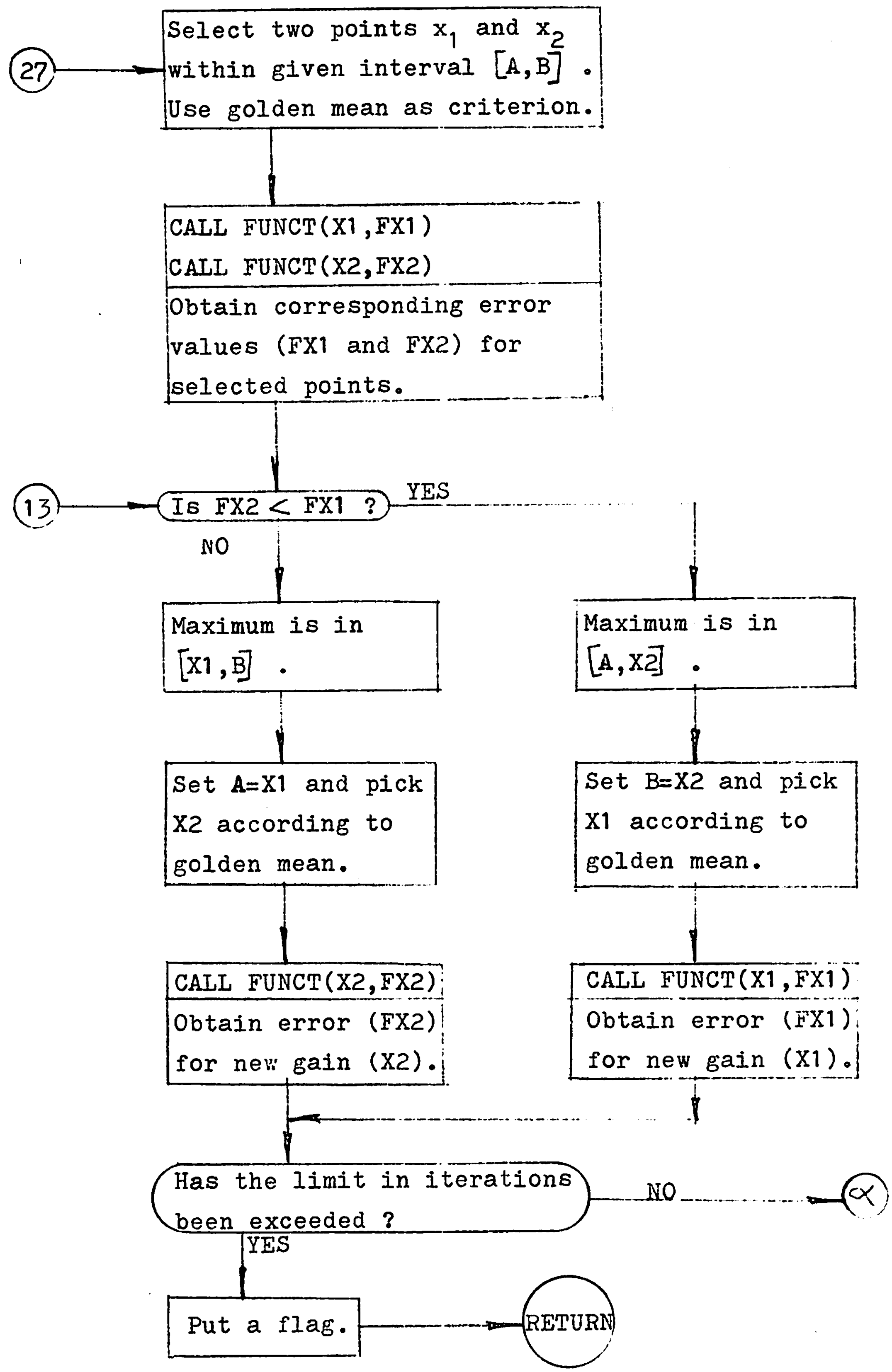
```
      DO 3002 J=1,MD
 3002 PRINT 56,J,VREF(J),ANGTER(J),CP(J),CQ(J),POWERS(J),VARS(J),VAMPS(J
     1),PTAR(J),QTAR(J),VATAR(J)
 56   FORMAT(I3,4X,F10.6,2X,F10.6,2X,F10.6,2X,F10.6,2X,F10.4,2X,F10.4,2X
     1,F10.4,2X,F10.4,2X,F10.4,2X,F10.4)
      PRINT 3003
 3003 FORMAT(/)
C     ******************************************************************
C     *PRINT INITIAL ERROR IN RMS PER UNIT VALUE.                      *
C     ******************************************************************
      PRINT 5,EPU
      CALL SECOND(PT2)
      ICOUNT=0
      IE=0
      EPUS=0.
C     ******************************************************************
C     *TAKE ONLY (2*MD-1) ELEMENTS IN X BECAUSE OF REFERENCE ANGLE.    *
C     ******************************************************************
      N=N-1
      CALL SECOND(T2)
      TIME=T2-T1-PT2+PT1
      PRINT 31,TIME
      PRINT 4008
 7    CALL SECOND(T1)
C     ******************************************************************
C     *OBTAIN THE GRADIENT OF ERROR FUNCTION.                          *
C     ******************************************************************
      CALL GRAD1
      ICOUNT=ICOUNT+1
C     ******************************************************************
C     *USE SECOND ORDER MOVE VECTOR TECHNIQUE ONLY.                    *
C     *COMPUTE MATRIX OF SECOND PARTIAL DERIVATIVES OF ERROR FUNCTION  *
C     *WITH RESPECT TO ADJUSTABLES.                                    *
C     ******************************************************************
      CALL ESPD(K)
      DO 8001 I=1,N
 8001 BB(I,1)=1.0
C     ******************************************************************
C     *INVERT MATRIX OF SECOND PARTIAL DERIVATIVES.                    *
C     ******************************************************************
      CALL MTINV1(BB,0,D,IRROR,IPIV,IND)
C     ******************************************************************
C     *CHECK IF IT IS POSSIBLE TO USE THE SECOND ORDER MOVE VECTOR. OB- *
C     *TAIN VECTOR OF CORRECTIONS.                                     *
C     ******************************************************************
      CALL METHOD(IRROR,ICOUNT)
C     ******************************************************************
C     *PERFORM A ONE-DIMENSIONAL SEARCH TO OBTAIN OPTIMAL LAMBDA.      *
C     ******************************************************************
      CALL INTER(AA,BB)
      CALL ODS(AA,BB,TOLER,100,ALAMB,FMAX)
C     ******************************************************************
C     *OBTAIN NEGATIVE OF CORRECTIONS AND CORRECT VECTOR X.            *
C     ******************************************************************
      DO 20 J=1,N
```

```
      D=ALAMB*DELX(J)
      X(J)=X(J)-D
 20   CONTINUE
C    *****************************************************************
C    *COMPUTE NEW ERROR.                                              *
C    *****************************************************************
      CALL REST(ENORMSQ)
      ENA=2.*MD
      EPU=SQRT(ENORMSQ/ENA)
C    *****************************************************************
C    *RESTITUTE ENTIRE DIMENSION IN X FOR PRINTING PURPOSES.          *
C    *****************************************************************
      N=NN
C    *****************************************************************
C    *CHECK IF THE DIFFERENCE BETWEEN NEW AND PAST ERROR IS LESS THAN *
C    *OR EQUAL TO E-10 FOR TWO CONSECUTIVE TIMES.                     *
C    *****************************************************************
      DEPU=ABS(EPU-EPUS)
      IF(DEPU.LE.1.0E-10)11,13
 11   IE=IE+1
      IF(IE.GE.2)GO TO 10
      IF(IE.EQ.1)GO TO 12
 13   IE=0
 12   EPUS=EPU
C    *****************************************************************
C    *CHECK IF THE VALUE OF THE ERROR IS LESS THAN OR EQUAL TO THE DESI-*
C    *RED MINIMUM.                                                    *
C    *****************************************************************
      IF(ABS(EPU).LE.EPSI)GO TO 10
C    *****************************************************************
C    *CHECK NUMBER OF ITERATIONS.                                     *
C    *****************************************************************
      M=M-1
      IF(M.LE.0) GO TO 35
      CALL SECOND(T2)
C    *****************************************************************
C    *PRINT NEW RMS PER UNIT ERROR.                                   *
C    *****************************************************************
      PRINT 9,ICOUNT
      PRINT 5,EPU
      TIME=TIME+T2-T1
      PRINT 30,TIME
      PRINT 4008
 4008 FORMAT(//)
C    *****************************************************************
C    *SET DIMENSION OF X TO N-1 AND PERFORM ANOTHER ITERATION.        *
C    *****************************************************************
      N=N-1
      GO TO 7
   35 PRINT 6, ALAMB
    6 FORMAT(* MORE ITERATIONS NEEDED LAMBDA=*,F10.4)
   10 CONTINUE
      CALL SECOND(T2)
      TIME=TIME+T2-T1
      AVEANG=0.
```

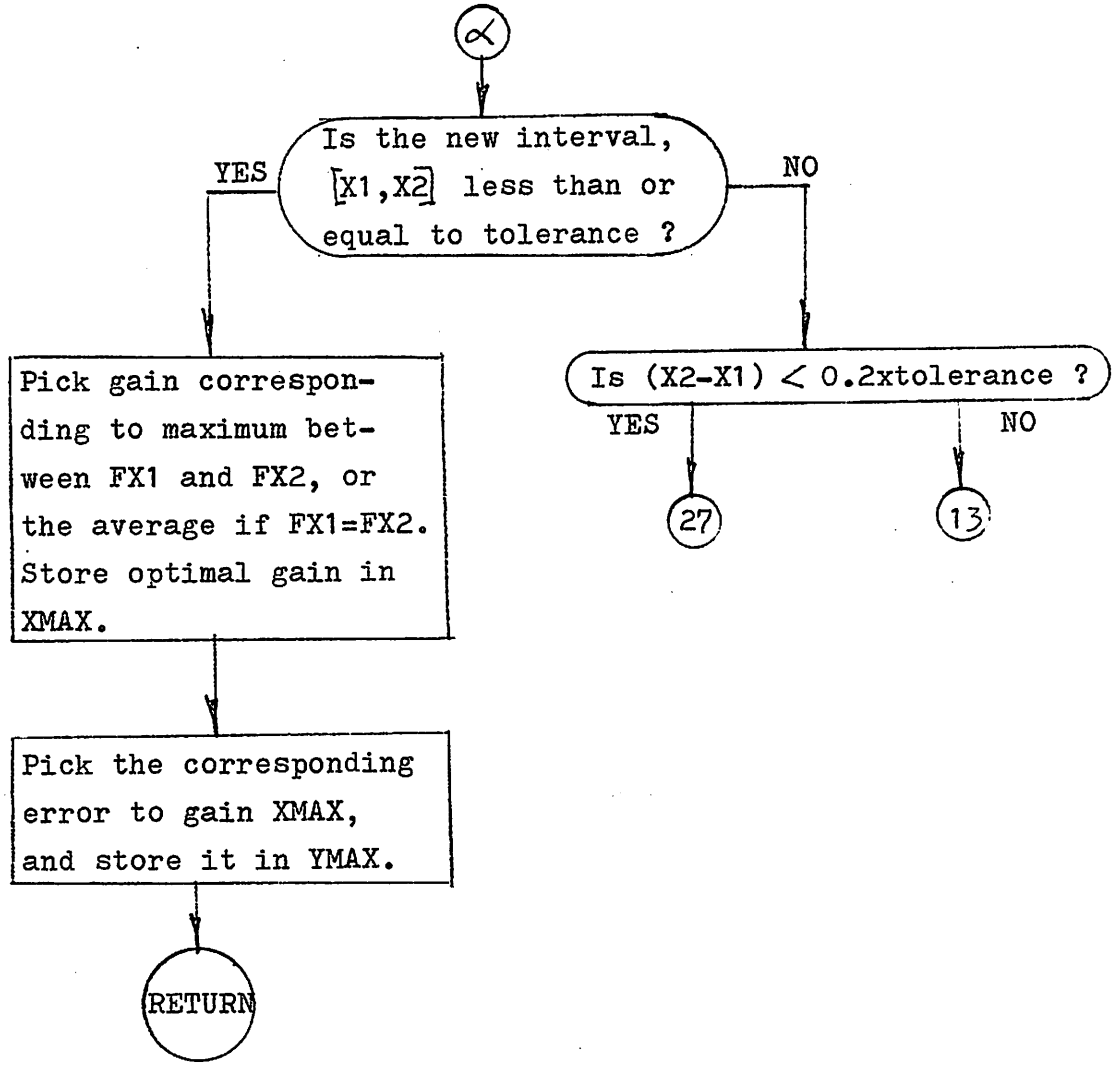
```
C     ******************************************************************
C     *COMPUTE EXCITATION VOLTAGES AND EXCITATION ANGLES.               *
C     ******************************************************************
      DO 52 J=1,NM
      VMR=ZMR(J)*UMR(J)-ZMI(J)*UMI(J)+VREF(J)*COS(ANGTER(J))
      VMI=ZMI(J)*UMR(J)+ZMR(J)*UMI(J)+VREF(J)*SIN(ANGTER(J))
      VM(J)=SQRT(VMR**2+VMI**2)
      ANGMACH(J)=ATAN2(VMI,VMR)
C     ******************************************************************
C     *CALCULATE AVERAGE ANGLE.                                         *
C     ******************************************************************
 52   AVEANG=AVEANG+ROIN(J)*ANGMACH(J)/SUMIN
C     ******************************************************************
C     *CORRECT TERMINAL AND EXCITATION ANGLES.                          *
C     ******************************************************************
      DO 59 J=1,MD
 59   ANGTER(J)=ANGTER(J)-AVEANG+0.3
      DO 60 J=1,NM
 60   ANGMACH(J)=ANGMACH(J)-AVEANG+0.3
      PRINT 3006
 3006 FORMAT(53X,* OPERATING STATF*/)
      PRINT 53
 53   FORMAT(* NODE   TERM VOLTS   TERM ANG   EXC VOLTS    EXC ANGLE
     1 POWERS       VARS         VAMPS        PTAR         QTAR        VATA
     1R*)
C     ******************************************************************
C     *PRINT OPERATING STATE.                                           *
C     ******************************************************************
      DO 54 J=1,MD
      PRINT 56,J,VREF(J),ANGTER(J),VM(J),ANGMACH(J),POWERS(J),VARS(J),VA
     1MPS(J),PTAR(J),QTAR(J),VATAR(J)
 54   CONTINUE
      PRINT 3003
      PRINT 9,ICOUNT
      PRINT 2005,EPU
      PRINT 30,TIME
C     ******************************************************************
C     *CALCULATE LOSSES.                                                *
C     ******************************************************************
      DO 70 J=1,MD
      DO 70 K=1,MD
 70   TOTALPO=TOTALPO+RNR(K,J)*VREF(J)**2
      SUMLOSS=TOTALPI-TOTALPO
      PRINT 57,AVEANG
 57   FORMAT(* AVERAGE ROTOR ANGLE =*,F10.6)
      PRINT 66,TOTALPI
 66   FORMAT(* SUM OF POWER INPUTS =*,F15.6)
      PRINT 67,TOTALPO
 67   FORMAT(* SUM OF POWER OUTPUTS =*,F15.6)
      PRINT 58,SUMLOSS
 58   FORMAT(* SUM OF POWER LOSSES =*,F15.6)
C     ******************************************************************
C     *CHECK IF THERE IS ANOTHER ERROR FUNCTION TO USE.                 *
C     ******************************************************************
      IF(INDEX.LT.NF) GO TO 25
```

```
5     FORMAT(* EPU =*,F10.5)
2005  FORMAT(* MIN EPU =*,F10.5)
9     FORMAT(* ITERATION NUMBER*,I3)
30    FORMAT(* TIME AT THIS ITERATION =*,F10.5)
31    FORMAT(* TIME SPENT BEFORE ITERATING =*,F10.5)
      RETURN
      END
```

SUBROUTINE ODS(A,B,TOLER,L,XMAX,YMAX)*

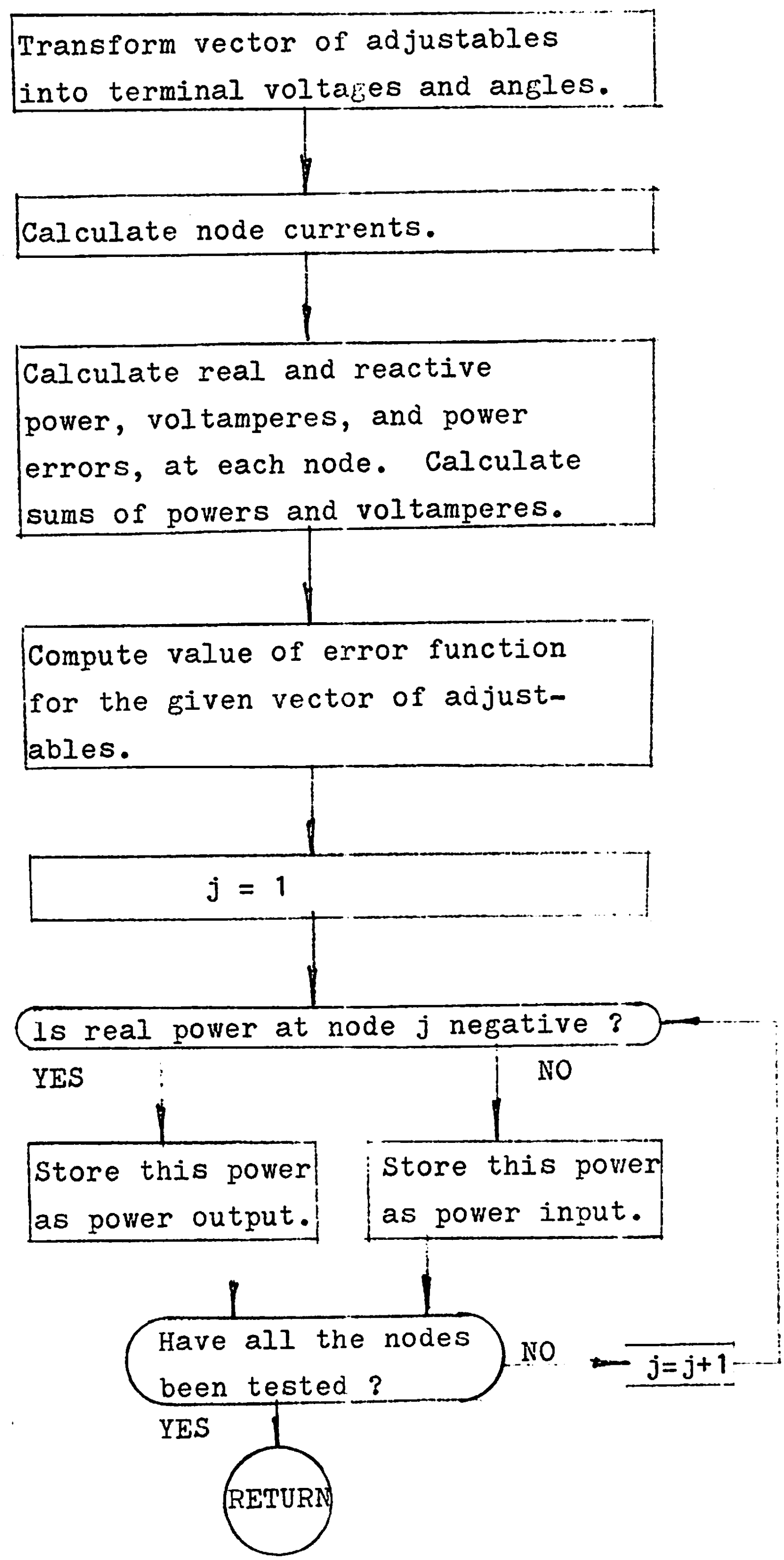


* See footnote in subroutine FUNCT


α
Is the new interval, [X1,X2] less than or equal to tolerance ?
YES
NO
Pick gain correspon-ding to maximum bet-ween FX1 and FX2, or the average if FX1=FX2. Store optimal gain in XMAX.
Pick the corresponding error to gain XMAX, and store it in YMAX.
RETURN
Is (X2-X1) < 0.2xtolerance ?
YES
NO
27
13

```
      SUBROUTINE ODS(A,B,TOLER,L,XMAX,YMAX)
C     ******************************************************************
C     *THIS SUBROUTINE PERFORMS A ONE DIMENSIONAL SEARCH FOR THE OPTIMAL *
C     *GAIN LAMBDA. THE MINIMUM OF THE FUNCTION ERROR VS. LAMBDA IS LOCA-*
C     *TED WITHIN THE TOLERANCE SPECIFIED IN THE INPUT. THE OUTPUT OF THE*
C     *SUBROUTINE IS THE OPTIMAL GAIN STORED IN XMAX, AND THE MINIMUM    *
C     *ERROR FOR THAT GAIN IS STORED IN YMAX.                            *
C     ******************************************************************
      COMMON INDEX
      II=0
   27 X2=(B-A)/(0.5*(1.+SQRT(5.))) + A
      X1=B+A-X2
      CALL FUNCT(X1,FX1)
   11 CALL FUNCT(X2,FX2)
      GO TO 13
   12 IF(X2-X1.LE.0.2*TOLER) GO TO 27
   13 IF(FX2.LT.FX1) GO TO 40
      IF(FX2.LT.FX1) GO TO 40
      A=X1
      X1=X2
      FX1=FX2
      X2=A+B-X1
      CALL FUNCT(X2,FX2)
      II=II+1
      EP=B-A
      IF(EP.LE.TOLER)GO TO 41
      IF(II.EQ.L) GO TO 42
      GO TO 12
   40 B=X2
      X2=X1
      FX2=FX1
      X1=B+A-X2
      CALL FUNCT(X1,FX1)
      EP=B-A
      IF(EP.LE.TOLER)GO TO 41
      II=II+1
      IF(II.EQ.L)GO TO 42
      GO TO 12
   42 EP=B-A
      PRINT 7,N,EP
    7 FORMAT (* NUM ITERATIONS=*,I6,* TOLERANCE=*,F10.6)
   41 CONTINUE
      IF (FX1.GT.FX2) X=A
      IF (FX1.LT.FX2) X=B
      IF(FX1.EQ.FX2) X=(A+B)/2.
      CALL FUNCT(X,FX)
      YMAX=AMAX1(FX,FX1,FX2)
      XMAX=X2
      IF(YMAX.EQ.FX) XMAX=X
      IF(YMAX.EQ.FX1) XMAX=X1
      RETURN
      END
```

SUBROUTINE REST(VALUE)



```
      SUBROUTINE REST(VALUE)
C     ******************************************************************
C     *THIS SUBROUTINE COMPUTES THE VALUE OF THE ERROR FUNCTION FOR A GI-*
C     *VEN VECTOR X. THE RESULT IS STORED IN VALUE. ALSO THE VECTOR X IS *
C     *DECOMPOSED IN VOLTAGES AND ANGLES AND STORED IN THE COMMON BLOCK  *
C     *REF. THE REAL AND IMAGINARY PARTS OF THE NODE VOLTAGES AND CURREN-*
C     *TS ARE STORED IN THIS BLOCK ALSO.                                 *
C     ******************************************************************
      COMMON INDEX
      COMMON/SHARE/X(100),DEL(100),A(100,100),N
      COMMON/CRST/DELX(100),DELXO(100)
      COMMON/ADMIT/RNR(8,8),RNI(8,8),MD
      COMMON/COSTS/CP2(8),CQ2(8)
      COMMON/TAR/PTAR(8),QTAR(8),VATAR(8)
      COMMON/DERIV/DPDV(8,8),DPDA(8,8),DQDV(8,8),DQDA(8,8)
      COMMON/REF/VREF(8),ANGTER(8),VR(8),VI(8),UMR(8),UMI(8)
      COMMON/POWER/TOTALPO,TOTALPI,POWERS(8),VARS(8),VAMPS(8)
      COMMON/SUMS/TOTALP,TOTALQ,TOTALVA
      DIMENSION PE(8),QE(8)
      DO 20 J=1,MD
      VREF(J)=X(J)
      JA=J+MD
   20 ANGTER(J)=X(JA)
      DO 206 J=1,MD
      VR(J)=VREF(J)*COS(ANGTER(J))
  206 VI(J)=VREF(J)*SIN(ANGTER(J))
      DO 207 J=1,MD
      UMR(J)=0
      UMI(J)=0
      DO 207 K=1,MD
      UMR(J)=UMR(J)+RNR(J,K)*VR(K)-RNI(J,K)*VI(K)
  207 UMI(J)=UMI(J)+RNR(J,K)*VI(K)+RNI(J,K)*VR(K)
      VALUE=0
      TOTALP=0.
      TOTALQ=0.
      TOTALVA=0.
      TOTALPO=0.
      TOTALPI=0.
      DO 511 J=1,MD
      VAR=VI(J)*UMR(J)-VR(J)*UMI(J)
      TOTALQ=TOTALQ+VAR
      VARS(J)=VAR
      QE(J)=VAR-QTAR(J)
      POWER=VR(J)*UMR(J)+VI(J)*UMI(J)
      TOTALP=TOTALP+POWER
      POWERS(J)=POWER
      VAMPS(J)=SQRT(POWER**2+VAR**2)
      TOTALVA=TOTALVA+VAMPS(J)
      PE(J)=POWER-PTAR(J)
      VALUE=VALUE+(CP2(J)*PE(J)**2+CQ2(J)*QE(J)**2)/VATAR(J)**2
      IF(POWER.LT.0)21,22
   21 TOTALPO=TOTALPO-POWER
      GO TO 511
   22 TOTALPI=TOTALPI+POWER
  511 CONTINUE
```

A.5 The output from program NNEWT2.

ADMITTANCE MATRIX OF NODES

	1		2		3		4	
1	14.92	-88.60	-2.19	23.90	-5.59	65.31	0.00	0.00
2	-2.19	23.90	18.91	-160.57	-13.38	137.59	0.00	0.00
3	-5.59	65.31	-13.38	137.59	78.01	-256.59	0.00	0.00
4	0.00	0.00	0.00	0.00	0.00	0.00	24.88	-213.47
5	0.00	0.00	0.00	0.00	0.00	0.00	-14.97	115.71
6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	0.00	0.00	0.00	0.00	-6.59	57.04	-9.90	99.01
8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

	5		6		7		8	
1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3	0.00	0.00	0.00	0.00	-6.59	57.04	0.00	0.00
4	-14.97	115.71	0.00	0.00	-9.90	99.01	0.00	0.00
5	25.91	-186.00	0.00	0.00	0.00	0.00	0.00	0.00
6	0.00	0.00	13.90	-182.17	0.00	0.00	-9.40	66.23
7	0.00	0.00	0.00	0.00	16.50	-154.12	0.00	0.00
8	0.00	0.00	-9.40	66.23	0.00	0.00	13.61	-140.18
9	0.00	0.00	0.00	113.64	0.00	0.00	-2.45	59.07
10	-10.93	71.32	0.00	0.00	0.00	0.00	-1.77	19.52

	9		10	
1	0.00	0.00	0.00	0.00
2	0.00	0.00	0.00	0.00
3	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00
5	0.00	0.00	-10.93	71.32
6	0.00	113.64	0.00	0.00
7	0.00	0.00	0.00	0.00
8	-2.45	59.07	-1.77	19.52
9	2.45	-169.41	0.00	0.00
10	0.00	0.00	12.70	-89.57

INVERSE NODE ADMITTANCE MATRIX

	1		2		3		4	
1	.014139	.007538	.012761	-.001538	.012498	-.002789	.012648	-.008061
2	.012761	-.001538	.013428	.003840	.012697	-.002111	.013139	-.007394
3	.012498	-.002789	.012697	-.002111	.012732	-.001670	.013353	-.006922
4	.012648	-.008061	.013139	-.007394	.013353	-.006922	.025932	.015576
5	.012417	-.009522	.012987	-.008872	.013252	-.008401	.028360	.013844
6	.008629	-.019335	.009727	-.018931	.010345	-.018542	.042665	-.001824
7	.012722	-.006225	.013112	-.005547	.013262	-.005082	.021685	.007361
8	.009491	-.019067	.010576	-.018614	.011182	-.018197	.042956	-.000028
9	.009030	-.019613	.010204	-.019184	.010830	-.018777	.043580	-.001242
10	.011833	-.011712	.012522	-.011103	.012864	-.010642	.031788	.010819

	5		6		7		8	
1	.012417	-.009522	.008629	-.019335	.012722	-.006225	.009491	-.019067
2	.012987	-.008872	.009727	-.018931	.013112	-.005547	.010576	-.018614
3	.013252	-.008401	.010345	-.018542	.013262	-.005082	.011182	-.018197
4	.028360	.013844	.042665	-.001824	.021685	.007361	.042956	-.000028
5	.032538	.020560	.052086	.003406	.023203	.005690	.052202	.005626
6	.052086	.003406	.116882	.048695	.031271	-.008197	.114482	.044530
7	.023203	.005690	.031271	-.008197	.019569	.009202	.031774	-.006915
8	.052202	.005626	.114482	.044530	.031774	-.006915	.113170	.049597
9	.053121	.004233	.118374	.048135	.032041	-.007914	.116219	.047120
10	.037203	.017335	.066771	.011568	.025253	.002901	.066616	.014451

	9		10	
1	.009030	-.019613	.011833	-.011712
2	.010204	-.019184	.012522	-.011103
3	.010830	-.018777	.012864	-.010642
4	.043580	-.001242	.031788	.010819
5	.053121	.004233	.037203	.017335
6	.118374	.048135	.066771	.011568
7	.032041	-.007914	.025253	.002901
8	.116219	.047120	.066616	.014451
9	.120036	.054567	.067993	.012778
10	.067993	.012778	.045640	.027500

IMPEDANCE MATRIX ON REGULATED VOLTAGE NODES

	1		2		3		4	
1	.014139	.007538	.012761	-.001538	.012498	-.002789	.012648	-.008061
2	.012761	-.001538	.013428	.003840	.012697	-.002111	.013139	-.007394
3	.012498	-.002789	.012697	-.002111	.012732	-.001670	.013353	-.006922
4	.012648	-.008061	.013139	-.007394	.013353	-.006922	.025932	.015576
5	.012417	-.009522	.012987	-.008872	.013252	-.008401	.028360	.013844
6	.008629	-.019335	.009727	-.019931	.010345	-.018542	.042665	-.001824
7	.012722	-.006225	.013112	-.005547	.013262	-.005082	.021685	.007361
8	.009491	-.019067	.010576	-.018614	.011182	-.018197	.042956	-.000028

	5		6		7		8	
1	.012417	-.009522	.008629	-.019335	.012722	-.006225	.009491	-.019067
2	.012987	-.008872	.009727	-.019931	.013112	-.005547	.010576	-.018614
3	.013252	-.008401	.010345	-.018542	.013262	-.005082	.011182	-.018197
4	.028360	.013844	.042665	-.001824	.021685	.007361	.042956	-.000028
5	.032639	.020560	.052086	.003406	.023203	.005690	.052202	.005626
6	.052086	.003406	.116862	.048695	.031271	-.008197	.114482	.044530
7	.023203	.005690	.031271	-.008197	.019569	.009202	.031774	-.006915
8	.052202	.005626	.114482	.044530	.031774	-.006915	.113170	.049597

ADMITTANCE MATRIX ON REGULATED VOLTAGE NODES

	1		2		3		4	
1	14.22	-98.60	-2.19	23.90	-5.59	65.31	.00	.00
2	-2.19	23.90	18.91	-160.57	-13.38	137.59	.00	.00
3	-5.59	65.31	-13.38	137.59	78.01	-256.59	-.00	-.00
4	.00	.00	-.00	.00	.00	.00	24.88	-213.47
5	-.00	-.00	.00	-.00	-.00	.00	-14.97	115.71
6	-.00	-.00	-.00	-.00	.00	.00	-.00	-.00
7	-.00	-.00	-.00	.00	-6.59	57.04	-9.90	99.01
8	.00	.00	.00	.00	-.00	-.00	.00	.00

	5		6		7		8	
1	-.00	-.00	.00	.00	.00	.00	-.00	.00
2	-.00	-.00	.00	.00	-.00	.00	-.00	.00
3	.00	.00	-.00	.00	-6.59	57.04	.00	.00
4	-14.97	115.71	-.00	.00	-9.90	99.01	.00	-.00
5	16.55	-129.22	-.00	-.00	.00	.00	-1.59	15.55
6	.00	.00	15.00	-105.96	.00	.00	-10.47	105.87
7	.00	.00	.00	.00	16.50	-154.12	-.00	-.00
8	-1.59	15.55	-10.47	105.87	-.00	-.00	12.03	-115.35

INTERNAL IMPEDANCE OF POWER NETWORK

	Z=ZCZ	R=ZCR	X=ZCI
1	.0111	.0018	.0110
2	.0062	.0007	.0061
3	.0037	.0011	.0036
4	.0047	.0005	.0046
5	.0077	.0010	.0076
6	.0093	.0013	.0093
7	.0065	.0007	.0064
8	.0086	.0009	.0086

NODE CONNECTION VECTORS OF TRANSMISSION LINES

LINE	1	2	3	4	5	6	7	8	9	10	11
NODE	3	2	3	7	7	5	10	10	1	1	2
NODE	1	1	2	3	4	4	5	8	8	9	6

MACHINE CHARACTERISTICS

NODE	INERTIA	RESISTANCE	REACTANCE	MAXIMUM POWER
1	5.710000	-0.000000	.114000	20.000000
2	8.710000	-0.000000	.078200	26.050000
3	2.640000	-0.000000	.447700	4.500000
4	7.110000	-0.000000	.063700	15.000000
5	8.090000	-0.000000	.031600	21.000000
6	10.530000	-0.000000	.074100	27.100000
7	-0.000000	-0.000000	-0.000000	0.000000
8	-0.000000	-0.000000	-0.000000	0.000000

INITIAL STATE

NODE	TERMINAL VOLTAGES		POWER ERROR WEIGHTS		COMPUTED POWERS			POWER TARGETS		
	TERM VOLTS	TERM ANG	CP	CQ	POWERS	VARS	VAMPS	PTAR	QTAR	VATAR
1	1.000000	-.300000	1.000000	1.000000	7.1500	-.6100	7.1760	19.2000	2.0000	19.3039
2	1.000000	-.300000	1.000000	1.000000	3.3500	-.9100	3.4714	23.0000	3.2000	23.2215
3	1.000000	-.300000	1.000000	1.000000	52.4500	-3.3500	52.5569	4.3000	.9500	4.4037
4	1.000000	-.300000	1.000000	1.000000	.0000	-1.2500	1.2500	15.0000	1.7500	15.1017
5	1.000000	-.300000	1.000000	1.000000	-.0114	-2.0428	2.0428	21.0000	2.5000	21.1483
6	1.000000	-.300000	1.000000	1.000000	4.5320	.0869	4.5328	26.0000	8.7000	27.4170
7	1.000000	-.300000	1.000000	1.000000	-.0000	-1.9300	1.9300	-4.9000	0.0000	4.9000
8	1.000000	-.300000	1.000000	1.000000	-.0171	-6.0760	6.0760	-31.5000	-11.5000	33.5336
TOTALS					67.4534	-16.0819	79.0359	72.1000	7.6000	149.0297

EPU = 2.81175

TIME SPENT BEFORE ITERATING = .01400

```
ITERATION NUMBER  8
MIN EPU =      .02359
TIME AT THIS ITERATION =     2.27800
AVERAGE ROTOR ANGLE  =          .300000

SUM OF POWER INPUTS  =     109.892404
SUM OF POWER OUTPUTS =     106.981123
SUM OF POWER LOSSES  =       2.911280
```

DISTRIBUTION LIST

Alekseev Aleksandr
Electromechanical Institute
Fontantka, 25,
Leningrad, U. S. S. R.

Mr. Robert Alexander
Power Authority of the State of New York
10 Columbus Circle
New York City, New York 10019

William G. Amey
Associate Director - Research Planning
Research and Development Center
Leeds and Northrup Company
Dickerson Road
North Wales, Pennsylvania 19454

Mr. Keith W. Amish
Chairman
Task Force on System Studies, NPCC
Rochester Gas & Electric Corporation
89 East Avenue
Rochester, New York 14604

Professor Michael Athans
Electrical Engineering Department
Massachusetts Institute of Technology
Cambridge, Massachusetts 02138

Dr. C. J. Bellamy
Director, Computer Centre
Monash University
Wellington Road, Clayton
Melbourne, Victori, Australia

Glenn W. Bills
Space Scientist
Research Department
Autonetics Division
North American Aviation, Inc.
12214 Lakewood Boulevard
Downey, California 90241

Dr. Robert R. Booth
System Planning Department
State Electricity Commission
of Victoria
22-32 William Street
Melbourne, Victoria, Australia

Lecturer William A. Brown
Electrical Engineering Dept.
Monash University
Wellington Road, Clayton
Melbourne, Victoria, Australia

J. A. Callow
System Design Engineer
Snowny Mountains Hydro-
Electric Authority
Cooma
New South Wales, Australia

John A. Casazza
System Planning and Develop-
ment Engineer
Public Service Electric and
Gas Company
80 Park Place
Newark, New Jersey 07102

Nathan Cohn, Senior Vice
President
Technical Affairs
Leeds and Northrup Company
4901 Stenton Avenue
Philadelphia, Penn. 19144

Charles Concordia, Consulting
Engineer
General Electric Company
Schenectady, New York

B. J. Cory
Electrical Engineering Department
Imperial College
University of London
London, England

H. W. Dommel
Bonneville Power Administration
U. S. Department of Interior
1002 NE Holladay
Portland, Oregon, 97232

Dr. James Eaton
International Business Machines Corp.
Monterey and Cottle Roads
San Jose, California 95114

Dr. Masakazu Ejiri
Central Research Laboratory - 6th Dept.
Hitachi, Ltd.
Kokubunji, Tokyo
Japan

Professor F. J. Evans
Senior Lecturer in Electrical Engineering
The University of Sydney
Sydney, New South Wales, Australia

Dennis Eyre
L. A. Department of Water & Power
P. O. Box 111
Los Angeles, California 90054

Professor A. E. Ferguson
Department of Electrical Engineering
University of Melbourne
Parkville, N. 2.,
Melbourne, Victoria, Australia

Roy H. Ferguson
Manager, Utility Systems
Canadian Westinghouse Co., Ltd.
286 Sanford Avenue
Hamilton, Ontario
Canada

Walter B. Fisk, Jr.
Consolidated Edison Company of New York
4 Irving Place
New York City, New York 10003

Mr. K. C. Fraser
Manager and Secretary
Electricity Commission of New South Wales
G. P. O. Box 5257
Sydney, N. S. W., Australia

Dr. S. G. Fraser
Electrical Engineering Dept.
University of Queensland
St. Lucia
Brisbane, Queensland, Australia

Mr. Gabor B. Furst
11 Orana Avenue
Cooma North, N. S. W., Australia

Mr. J. L. W. Harvey
System Planning Department
State Electricity Commission of Victoria
22-32 William Street
Melbourne, Victoria, Australia

Seymour W. Herwald
Group Vice-President
Westinghouse Electric Corp.
Gateway Center #3
401 Liberty Avenue
Pittsburgh, Pennsylvania 15222

Mr. Robert H. Hillery
Operating Procedure Coordinating Committee, NPCC
The Hydro-Electric Power Commission of Ontario
620 University Avenue
Toronto 2, Canada

G. S. Hope
Electrical Engineering Department
University of Calgary
Calgary, Canada

Professor R. M. Huey
Associate Professor of Electrical Engineering
The University of New South Wales
Box 1, Post Office, Kensington
Sydney, New South Wales, Australia

Seiji Inoue
Manager, Engineering Department
Electrical Machinery Division
Hitachi, Ltd.
Nippon Building
No. 8, 2-chome, Ohtemachi
Chiyoda-ku, Tokyo
Japan

Dr. Jacob N. Johnson
CEIR National Director for Power
Industry Analysis
5272 River Road
Washington, D.C. 20016

Mr. John Johnson
System Planning Department
State Electricity Commission of Victoria
22-32 William Street
Melbourne, Victoria, Australia

William R. Johnson
Chief, Electrical Generation and
Transmissions Engineers
Pacific Gas and Electric Company
245 Market Street
San Francisco, California 94105

G. Allan Jones
Planning Department
Pacific Gas and Electric Company
245 Market Street
San Francisco, California 94105

Lee Kilgore
Consulting Engineer
Westinghouse Electric Corp.
East Pittsburgh, Penn. 15112

K. Kumai
Kyushn Electric Power Co., Inc.
Fukuoka, Japan

Dr. Vno Lamm
1777 Borel Place
San Mateo, California 94402

Professor Douglas G. Lampard
Electrical Engineering Dept.
Monash University
Wellington Road, Clayton
Melbourne, Victoria, Australia

Dr. W. S. Leung
Electrical Engineering Dept.
Hong Kong University
Hong Kong

T. Lindstrom
Director of Research
Allmanna Svenska Elektriska
Aktiebolaget
Vasteras, Sweden

James Luini
Planning Department
Pacific Gas and Electric Co.
245 Market Street
San Francisco, California 94105

H. W. Lydick
District Engineer
Westinghouse Electric Corp.
One Maritime Plaza
San Francisco, California 94111

Professor C. E. Moorhouse
Department of Electrical Engineering
University of Melbourne
Parkville, N.2.,
Melbourne, Victoria, Australia

Mr. K. Morishita
Research Administration Department
Central Research Laboratories
Mitsubishi Electric Corporation
Amagasaki, Hyogo
Osaka, Japan

Lecturer Karol Morsztyn
Electrical Engineering Department
Monash University
Wellington Road, Clayton
Melbourne, Victoria, Australia

K. Ode
Control Research Institute of
Electric Power Industry
Tokyo, Japan

Professor Dr. Ing. Winfried Oppelt
Institut fur Regelungstechnik
der Technishen Hochschule Darmstadt
Darmstadt, West Germany

Tokuma Ozawa
Chief of Power System Engineering Section
Technical Department
Fuji Electric Company, Ltd.
Fawasaki, Japan

Dr. A. James Pennington
Electrical Engineering Department
University of Michigan
Ann Arbor, Michigan 48104

H. Ray Perry
Supervising Electrical Engineer
Pacific Gas and Electric Company
245 Market Street
San Francisco, California 94105

John Peschon
Senior Research Engineer
Information and Control Laboratory
Stanford Research Institute
Menlo Park, California, 94025

Harold A. Peterson
Electrical Engineering Department
University of Wisconsin
Madison, Wisconsin 53706

Mr. S. Saba
Manager, Electric Power
Engineering Department
Tokyo Shibaura Electric Co., Ltd.
1-1, Uchisaiwaicho
Chiyoda-Ku, Tokyo, Japan

Harvey Sahlin
Livermore Radiation Laboratory
University of California
Livermore, California 94550

Dr. Saito
Chief, Central Research Laboratory
Fuji Electric Company, Ltd.
Yokosuka
Kanaga Prefecture, Japan

Dr. Norikazu Sawazaki
Manager, Development Department
Central Research Laboratory
Tokyo Shibaura Electric Co., Ltd.
1, Toshiba-cho, Komukai
Kawasaki, Japan

Professor Toshio Sekiguchi
Electrical Engineering Department
Tokyo Institute of Technology
O-Okayama, Meguroku
Tokyo, Japan

Professor Yasiyi A. Sekine
Electrical Engineering Department
University of Tokyo, Hongo
Bunkyo-ku
Tokyo, Japan

Fred P. Sever
Senior Planning Engineer
Cleveland Electric Illuminating Co.
P. O. Box 5000
55 Public Square
Cleveland, Ohio 44113

Mr. H. Sho
Assistant Manager
Power Machinery Section B
Foreign Trade Department A
Mitsubishi Electric Corporation
Mitsubishi Denki Building
Marunouchi, Tokyo, Japan

Mr. Don C. Smith
System Planning Department
State Electricity Commission of Victoria
22-32 William Street
Melbourne, Victoria, Australia

N. A. Smith
Electrical Engineering Department
The Ohio State University
2024 Neil Avenue
Columbus, Ohio 43210

Professor Benam Speedy
Head of Control Department in
Electrical Engineering
University of New South Wales
Sydney, New South Wales, Australia

G. W. Stagg
Head, Engineering Analysis and Computer
Division
American Electric Power Service Corp.
2 Broadway
New York City, N. Y. 10004

State Electricity Commission of Queensland
447 Gregory Terrace
Brisbane, Queensland, Australia

K. Takahashi
Electrical Engineerring Department
University of Tokyo, Hongo
Bunkyo-ku
Tokyo, Japan

William F. Tinney, Head,
Methods Analysis Group
Bonneville Power Administration
U. S. Department of Interior
1002 NE Holladay
Portland,Oregon, 97232

Mr. Albert D. Tuttle, Chairman
Task Force on System Protection,
NPCC
New York State Electric & Gas Co.
4500 Vestal Parkway East
Binghamton, New York 13902

T. Umezu
Central Research Institute of
Electric Power Industry
1229 Iwato Komae-cho, Kitayama-
Guord, Bunkyo-ku
Tokyo, Japan

Professor James E. VanNess
Electrical Engineering Department
Northwestern University
Evanston, Illinois 60201

Professor A. A. Voronov
Institute of Automatics and Tele-
mechanics of the Academy of
Sciences of the U. S. S. R.
Kalanchevskaja, 15a
Moscow, 1-53, U. S. S. R.

J. Wilson
Engineer/Systems Planning
Electricity Commission of N. S. W.
Kelvin House
15 Castlereagh Street
G. P. O. Box 5257
Sydney, N. S. W., Australia

Ing. Jesus Lozano de los Santos
Division Ingenieria
Hojalata y Lamina, S. A.
Apartado No. 996
Monterrey, N. L., Mexico

Ing. J. A. Gonzalez Arechiga
ITESM
Departamento de Ingenieria Mecanica
Sucursal "4" de Correos
Monterrey, N. L., Mexico

Ing. Antonio Elizondo
ITESM
Departamento de Ingenieria Electrica
Sucursal "4" de Correos
Monterrey, N. L., Mexico

Ing. Antonio R. Zarate
Topo No. 308
Col. Chapultepec
Monterrey, N. L., Mexico

Professor Jens Balchen
Department of Electrical Engineering
University of California at Santa Barbara
Santa Barbara, California 93106