Load Flow as a Planning Tool - An Optimal Approach

By

R. Shanthi

M. Jayakumar

V. S. Ravichandran

G. Shanthanayaki

Shiny Purushothaman

Guided By

Er K. Surendran

Faculty of Electrical & Electronics Engineering
Kumaraguru College of Technology
Coimbatore - 641 006

Project Work
1989

Faculty of Electrical and Electronics Engineering

Kumaraguru College of Technology

Coimbatore - 641 006

Certificate

This is the Bonafide Record of the Project titled

Load Flow as a Planning Tool - An Optimal Approach

done by

Mr. / Miss. .... Roll No. ....

In partial fulfilment of the requirements for the Degree of
Bachelor of Engineering in Electrical and Electronics Engineering
Branch of Bharathiar University, Coimbatore-641 046
during the year 1988-89

Er. K. Surendran
(Project Guide)

Dr. K. A. Palaniswamy, Ph.D.
(Head of the Department)

Dr. K. A. PALANISWAMY, B.E.,M.Sc.(Engg).,Ph.D.,
MISTE.C,Engg (I).,FIE.,
Professor and Head
Department of Electrical and Electronics Engineering,
Kumaraguru College of Technology,
Coimbatore-641 006

Submitted for the University Examination held on ....

University Register No. ....

Internal Examiner

External Examiner

Dedicated
To
Our Parents

acknowledgement

ACKNOWLEDGEMENT

At the juncture of accomplishment of our project, we acknowledge with due courtesy **Mr. K. SURENDRAN M.Tech.,** for his guidance and inspiration. We are grateful to our Professor **Mr.K.A. PALANISWAMY** Ph. D, M.I.S.T.E., F.I.E., for his valuable suggestions and encouragement.

It is our privelege to thank Principal Professor, **R.M. LAKSHMANAN, B.E., M.Sc., (Engg), M.I.S.T.E., M.S.A., E.S.T.,** for his co-operation in our endeavour.

Our sincere thanks are due to Professor **P.SHANMUGAM, M.Sc., (Engg), M.S. (HOWAII) M.I.E.E.,** for the help rendered to us.

We are greatly indebted to Mr. **K.RAMAN NAIR** B.Sc., **(Engg) M.Tech.,** Ph.D., for suggesting us this project and for his constant help and guidance.

We extend our deep sense of gratitude to **Mr. R.P. THANGARAJ M.Sc., M.Phil.,** who helped us a lot in drafting the software techniques of our project.

A special word of thanks is, due to **Miss. MUTHAMANI M.E.,** for her encouragement throughout our project.

A final work of thanks is extended to the faculty members of Electrical and Electronics Department for their whole hearted co-operation in bringing this project into shape.

abstract

LOAD FLOW AS A PLANNING TOOL- AN OPTIMAL APPROACH

ABSTRACT:

The load flow solution is an indespensable tool in the design or operational planning of a power system. In practical systems, in order to obtain an acceptable design or operating strategy, a large number of conventional load flow solution have to be performed one after the other. The success of the final solution depends to a great extent on the experience, skill and intuition of the system planner and invariably involves trial and error adjustments.

In this Report, a new approach is conceived wherein a single optimal load flow solution replaces a series of many trial and error load flow solutions. For a system under given constraints the best possible operating strategy is directly found by using one line solution technique. This method is illustrated with an example. While the proposed technique calls for greater programing effort and computer memory compared to conventional load flows, this would not be a serious drawback with the availability of the present day powerful computing aids.

nomenclature

NOMENCLATURE

P - Real Power

Q - Reactive Power

V - Voltage magnitude

- Voltage angle in radians

N - Number of buses

G - Number of generator buses

SUPER SCRIPT

-1 - Inverse of the Matrix

tr - Transpose of the Matrix

contento

introduction

CONTENTS

INTRODUCTION

The load flow solution techniques based on Gauss Seidal or Newton Raphson Algorithms are so frequently used in the process of power system design or operational planning so that its basic features and limitations have been taken for granted. Each bus in a system is characterised by four variables, $|V|$ and δ at the slack bus, P and $|V|$ at generator buses and P and Q at the load buses. The load flow solution gives the corresponding values of P, and Q at the slack bus, Q and δ at the generator buses and $|V|$ and δ at the load buses. Often the results of the initial load flow are not acceptable. For example the P and Q found at the slack bus may be beyond the capacity of the generator at that bus. Similarly $|V|$ as found at some of the load buses may be too low. Or at a generator bus, due to the voilation of limits on Q, the voltage may be too high or too low. By reducing P at such a bus, the limits for Q may be widened and this could lead to an acceptable voltage level. Thus, the system planner has to consider several alterations, modify the input specifications to the load flow problem and obtain a fresh solution. Often a number of such load flows may be carried out before an acceptable result is obtained.

In this work, a new method is proposed which enable the system planner to specify desired values to any number of general

solution.

By this method the optimum results are achieved with a

optimization techniques

variables in the system. The load flow solution is reconceived as one that yields a system state that is closest to the desired solution. To solve the problem, an optimization technique based on weighted least square errors principle is used. Provision are also made for the system planner to incorporate the relative importance that he attaches to each of the various desired values. The method yields a solution in which the sum of the weighted squares of the deviation of each variable from its desired or target value is minimum subject to the necessary load flow constraints. without the need to conduct several case studies. Thus the process of arriving at the most acceptable system state is made more systematic and objective., and less time consuming, and the subjective decisions which the system planner normally has to make are almost entirely eliminated.

CHAPTER I

OPTIMIZATION TECHNIQUES:

Art or technique of obtaining best results under certain restrictions and limits is known as the optimization. The objective function may be different from problem to problem. The method for getting the results may also vary. In general the optimization techniques fall into two categories, viz Linear and Nonlinear techniques. Out of these two, linear methods are more effective, less time consuming and for real time implementation they are well suited. In this chapter weighted least square error method is revieved briefly. A feasible solution which optimizes the objective function is called an optimal solution.

weighted least square error algorithm

This result is also in accord with experience and common sense If the accidental errors of n measurments $m_1, m_2, \ldots\ldots m_n$ be denoted by $x_1, x_2, \ldots\ldots x_n$ respectively then

$$x_1^2 + x_2^2 + x_3^2 + \ldots\ldots + x_n^2 = \sum x^2 \text{ is a minimum.}$$

We arrive at this principle from the principle of probability, which states that the best or most probable value detainable from a set of measurements or observations of equal precesion is that value for which the sum of the squares of the errors is a minimum.

In its most general form, the above principle states that the best value of an unknown quantity that can be obtained from a set of measurements of unequal precision is that which makes the. sum of the weighted squares of the errors a minimum.

Adopting this principle to the optimal load flow problem, we define the best state of the power system as that for which the sum of the weighted squares of the derivations of each variable from its derived value is a minimum. The weightages al'ow the system planner to assign the relative importance of the various variables. This is in fact the only point at which the subjective assessment of the system planner can affect the solution.

[illegible] [illegible] direct measurements

problem formulation

CHAPTER III

PROBLEM FORMULATION:

Problem is formulated in polar co-ordinates. Variables at each bus are P, Q, V and δ

N → total number of buses

1 to G → Generator buses of which bus 1 is slack bus, so that $S_1 = 0$

G + 1 to N → load buses, at which P and Q are specified.

δi, i = 2 to N and V_i, i = 1 to N are the system state variables numbering 2N - 1 . There are 2N other variables, namely P_i , i = 1 to N and Q_i, i = 1 to N.

In conventional load flow studies one has to specify values for 2N of the 4N variables and then solve the non linear power equations for the remaining 2N variables. However in practice we really need to specify only the slack bus angle and P and Q for the load buses. All other specifications are artificially introduced.

Let there be G generator buses, including the slack bus. Then G+1, G+2,.........N are the load buses. For each of these buses, P_i and Q_i must be specified. So the total number of specified variables is K = 2(N-G).

From among the 2N-1 state variables, we choose K variabnles called dependent variables, forming a vector [x] of order K x 1.

$$[x] = \begin{array}{c} 1 \\ 2 \\ \\ \\ N-G \\ N-G+1 \\ \\ \\ K=2(N-G) \end{array} \left[\begin{array}{c} \delta_{G+1} \\ \delta_{G+1} \\ \vdots \\ \vdots \\ \delta_N \\ V_{G+1} \\ V_{G+2} \\ \vdots \\ V_N \end{array} \right]$$

The remaining 2N-1-k variables is δ_2 to δ_G, V_1 to V_G are called independent variables and form a vector [Y] of order j x 1 where j = 2N-1-k = 2G-1

$$[Y] = \begin{array}{c} 1 \\ 2 \\ \vdots \\ G-1 \\ G \\ G \\ \vdots \\ j=2G-1 \end{array} \left[\begin{array}{c} \delta_2 \\ \delta_3 \\ \vdots \\ \delta_G \\ \hline V_1 \\ V_2 \\ \vdots \\ V_G \end{array} \right]$$

$$[Q] = \begin{bmatrix} Q_1 \\ Q_2 \\ . \\ . \\ Q_G \end{bmatrix} \quad [Q_{min}] = \begin{bmatrix} Q_{1min} \\ Q_{2min} \\ . \\ . \\ Q_{Gmin} \end{bmatrix} \quad [Q_{max}] = \begin{bmatrix} Q_{1max} \\ Q_{2max} \\ . \\ . \\ Q_{Gmin} \end{bmatrix}$$

There is a third class of variables for whoch we want to have values as close as possible to give 'target' values while also lying between specified lower and upper limits. This includes P at each generator bus and V for all buses. These 'target' variables are included in a vector [b] of order m X 1 where m = N+G. Their target values are included in a vector [t] and their limits in vectors $[b_{min}]$ and $[b_{max}]$ all of order m x 1.

$$[b] = \begin{array}{c} 1 \\ 2 \\ . \\ . \\ G \\ \\ G+1 \\ G+2 \\ . \\ . \\ 2G \\ 2G+1 \\ . \\ . \\ m \end{array} \begin{bmatrix} P_1 \\ P_2 \\ . \\ . \\ P_G \\ \hdashline V_1 \\ V_2 \\ . \\ . \\ V_G \\ V_{G+1} \\ . \\ . \\ V_N \end{bmatrix} \qquad [b_{min}] = \begin{array}{c} 1 \\ 2 \\ \\ \\ G \\ \\ G+1 \\ \\ \\ \\ 2G \\ 2G+1 \\ . \\ . \\ m \end{array} \begin{bmatrix} P_{1min} \\ P_{2min} \\ . \\ . \\ P_{Gmin} \\ \hdashline V_{1min} \\ V_{2min} \\ . \\ . \\ V_{Gmin} \\ V_{G+1min} \\ . \\ . \\ V_{Nmin} \end{bmatrix}$$

$$[b_{max}] = \left[\begin{array}{l} P_{1max} \\ P_{2max} \\ . \\ . \\ P_{Gmax} \\ \hdashline V_{1max} \\ V_{2max} \\ . \\ . \\ V_{Gmax} \\ V_{G+1max} \\ . \\ . \\ V_{Nmax} \end{array}\right]$$

$$[t] = \left[\begin{array}{l} t_1 \\ t_2 \\ . \\ . \\ t_G \\ \hdashline t_{G+1} \\ t_{G+2} \\ . \\ . \\ t_{2G} \\ t_{2G+1} \\ . \\ . \\ t_m \end{array}\right]$$

All the specified variables are included in a vector [a] of order k x 1. The specified values for these variables are grouped in a vector [s] of order k x 1.

$$[a] = \begin{array}{c|c|} & P_{G+1} \\ 1 & \\ & P_{G+2} \\ 2 & \\ & \cdot \\ \cdot & \\ & \cdot \\ \cdot & \\ N-G & P_N \\ \hdashline N-G+1 & Q_{G+1} \\ \cdot & Q_{G+2} \\ & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ K & Q_N \end{array} \qquad [S] = \begin{bmatrix} S_1 \\ S_2 \\ \cdot \\ \cdot \\ S_{N-G} \\ \hdashline S_{N-G+1} \\ \cdot \\ \cdot \\ \cdot \\ S_K \end{bmatrix}$$

Each element in [a] is a function of the variables [x] and [y] . We may therefore write

$$[a] = [a\ \{(x),\ (y)\}\ ]$$

The load flow solution must satisfy the conditions

$$[a\ \{(x)\ ,\ (y)\}\ ] \qquad [s] \text{ -----(1)}$$

Now the Q at each generator bus is required to lie between lower and upper limits. The load flow solution must satisfy the conditions.

$$[Q_{min}] \leq [Q\ \ (x)\ ,\ (y)\ \ ] \leq [Q\ _{max}] \text{ ----(2)}$$

where [Q] , $[Q_{min}]$ and $[Q_{max}]$ are G x 1 vectors given by

The load flow solution must satisfy the condition

$$[b_{min}] \leq [b \; (x), \; (y) \;] \leq [b_{max}] \qquad (3)$$

and

$$[b \; \{(x) \; , \; (y)\} \;] \text{ must lie as close as possible to } [t] \qquad \text{-----}(4)$$

The heuristic condition of equation (4) may be written as a formal mathematical weighted least squares proportion, namely

$$F = \sum_{i=1}^{m} W_i \; (t_i - b_i)^2 \text{ must be minimum.}$$

$F = (\; [t] - [b]^{tr} \; [W] \,)([t] - [b] \;)$ must be minimum where $[W]$ is a m x m diagonal matrix whose diagonal elements are the weightages w_i

$$[W] = \begin{bmatrix} W_i & 0 \; .... \; 0 \\ 0 & W_2 .... 0 \\ \vdots & \\ 0 & 0 & W_m \end{bmatrix}$$

The load flow problem may now be formulated to an optimisation problem, as given below. Minimise

$$F = (\; [t] - [b\{ \; [x] \; [y]\} \;]^{tt} \; [W] \,)([t] - [b\,\{[x],[y]\} \;] \;) \quad \text{---} \; (5)$$

Subject to the constraints

$$[a \{ [x], \; [y]\} \;] \; = [s] \qquad \text{------}(1)$$

$$[Q_{min}] \leq [Q \; \{[x], \; [y] \; \} \;] \leq [Q_{max}] \qquad \text{-----}(2)$$

$$[b_{min}] \leq [b \; \{ [x], \; [y] \} \;] \leq [b_{max}] \qquad \text{-----}(3)$$

Note that $K = 2(N-G) = 2N - 2G$

and so $K < 2N - 1$ or $K <$ no of state variables.

Also, $m = N+G = j + N + G + 1$ and so $m \quad j$ or $m >$ number of independent variables. Since the number of specification K is less than the number of state variables 2N-1, an infinite number of solution exiist which will satisfy equation 1. Out of these, we are looking for that solution which will minimise F of equation (5) and also satsify and also satisfy the constraints of equation (2) and (3).

derivation of the algorithm

$= ([t] - [b] - [c] [\Delta x] - [D] [\Delta y])^{tr} [W] ([t] - [b] - [c] [\Delta x] - [D][\Delta Y])$

where $[c] = [db/dx]$ is a $m \times k$ matrix

and $[D] = [db/dy]$ is a $m \times j$ matrix

$= ([t] - [b] - [c] [A]^{-1} ([s] - [a]) - ([D] - [c] [A]^{-1} [B] [\Delta Y])^{tr} [W] ([t] - [b] - [c] [A]^{-1} ([s] - [a]) - ([D] - [c] [A]^{-1} [B]) [\Delta y]$ when substituting for $[\Delta x]$ from equation (6)

$= ([t] - [b] - [c] [A]^{-1} ([s] - [a]) - [G] [\Delta y])^{tr} [w] ([t] -[b] - [c] [A]^{-1} ([s] - [a]) - [G] [\Delta y])$

where $[G[= [D] - [C] [A]^{-1} [B]$ is a $m \times j$ matrix.

$= ([t] - [b] - [c] [A]^{-1} ([s] - [a]))^{tr} [w] ([t] - [b] - [c] [A]^{-1} ([s] - [a])) - 2 ([t] - [b] - [c] [A]^{-1} ([s] - [a]))^{tr} [W] [G] [\Delta y] + ([G] [\Delta y])^{tr} [W] ([G] [\Delta y])$ must be a minimum

therefore $dF/\Delta Y$ must be zero ie

$$-2 ([t] - [b] - [c] [A]^{-1} ([s] - [a]))^{tr} [W] [G] + 2([G] [\Delta y]^{tr} [W] [G] = 0$$

transposing

$$-2[G]^{tr} [W] ([t] - [b] - [c] [A]^{-1} ([s] - [a])) + 2[G]^{tr} [W] ([G][\Delta y]) = 0$$

ie

$$[G]^{tr} [W] [G] [\Delta Y] = [G]^{tr} [W] ([t] - [b] - [c] [A] ([s] - [a]))$$

or

$$[\Delta y] = ([G]^{tr} [W] [G])^{-1} [G]^{tr} [W] ([t] - [b] - [c] [A]^{-1}([s]-[a]) \qquad 7$$

CHAPTER IV

DERIVATION OF THE ALGORITHM:

Since the load flow problem is non-linear an iterative procedure must be used. At any iteration, let [x] and [y] represent the values of the state variables. Let $[\Delta x]$, $[\Delta y]$ be the optimal corrections to be made to [x] and [y] to proceed to the next iteration.

Then since we are aiming to satisfy equation 1,

$[S] - [a\{ [x] + [\Delta x], [y] + [\Delta y]\}] = 0$

Let $[a\{[x] + [\Delta x] , [y] + [\Delta y]\}] = [a\{ [x],[y] \}] + [\Delta a]$

Then $[s] - (a \{[x], [y]\}] + [\Delta a]) = 0$

ie $[S] - [a] - [\Delta a] = 0$

ie $[s] - [a] - [A]\Delta x - [B] \Delta y = 0$

where $[A] = [da/dx]$ is a k x k matrix

and $[B] = [da/dy]$ is a k x j matrix

ie $[A] [\Delta x] = [s] - [a] - [B] [\Delta y]$

pr $[\Delta x] = [A]^{-1} ([s] - [a] - [B] [\Delta y])$ -----(6)

Also since we are assuming to minimise F,

from equation 5,

$F = ([t] - [b \{[x] + [\Delta x], [y] + [\Delta y]\}] = [b\{ [x],[y]\}]+[\Delta b]$

must be a minimum

let $[b\{ [x] + [\Delta x], [y] + [\Delta y]\}] = [b \{ [x],[y]\}] + [\Delta b]$

then $F = ([t] - [b \{[x],[y]\}] -[\Delta b])^{tr} [W] ([t] - [b\{ [x],[y]\}] - [\Delta b])$

iterative procedure

CHAPTER V

ITERATIVE PROCEDURE:

The starting values for the state variables [x] and [y] may be chosen as follows:

$$[x]_0 = \begin{bmatrix} \delta_{G+1} \\ \delta_{G+2} \\ \vdots \\ \delta_N \\ \hline V_{G+1} \\ V_{G+2} \\ \vdots \\ V_N \end{bmatrix} = \begin{bmatrix} o \\ 0 \\ \vdots \\ 0 \\ \hline 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} ; \quad [y]_0 = \begin{bmatrix} \delta_2 \\ \delta_3 \\ \vdots \\ \delta_G \\ \hline V_1 \\ V_2 \\ \vdots \\ V_G \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ \hline 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$$

Let the state variables at the end of an iteration be [x], [y] . To proceed to the next iteration:-

Step 1

Calculate P and Q for all the buses and form the vectors [a], [Q] and [b].

Step 2

The MVA rating of each generator bus is known, say S_i for the ith bus. Calculate $Q_{mini} = -\sqrt{S_i^2 - P_i^2}$ and $Q_{maxi} = \sqrt{S_i^2 - P_i^2}$

$$\underset{K' \times 1}{[\ a'\ ]} = \begin{bmatrix} P_{G+1} \\ P_{G+2} \\ \vdots \\ P_N \\ ---- \\ Q_{G+1} \\ Q_{G+2} \\ \vdots \\ Q_N \\ ----- \\ Q_2 \\ Q_5 \end{bmatrix} \qquad \underset{K' \times 1}{[\ S'\ ]} = \begin{bmatrix} S_1 \\ S_2 \\ \vdots \\ S_{N-G} \\ ----- \\ S_{N-G+1} \\ S_{N-G+2} \\ \vdots \\ S_K \\ ----- \\ Q_{2min} \\ Q_{5\,max} \end{bmatrix}$$

If the Q' s for all the generator buses are within this limits, simply set $K' = K$, $j' = j$, $[\ x'\ ] = [\ x\ ]$, $[\ y'\ ] = [\ y\ ]$, $[\ a'\ ] = [\ a\ ]$ and $[\ s'\ ] = [\ s\ ]$.

Step 3

Check each variable in [b] for limit violation. If it is found that $b_i < b_{mini}$ or $b_i > b_{maxi}$ increase W_i suitably and thus form a new [W]

Step 4

Compute $[\ s'\ ] - [\ a'\ ]$ and store it is in the same location as $[\ a'\ ]$, compute $[\ t\ ] - [\ b\ ]$ and store it in the location as $[\ b\ ]$, compute the $K' \times K'$ matrix $[\ A'\ ]$ as follows.

$$
[A'] = \begin{array}{|cccccccccc|l}
\frac{\delta P_{G+1}}{\partial \delta_{G+1}} & \frac{\delta P_{G+1}}{\partial \delta_{G+2}} & \cdots \frac{\delta P_{G+1}}{\partial \delta_{N}} & & \frac{\partial P_{G+1}}{\partial V_{G+1}} & \frac{\delta P_{G+1}}{\partial V_{G+2}} & \cdots & \frac{\delta P_{G+1}}{\partial V_{N}} & \frac{\delta P_{G+1}}{\partial V_{2}} & \frac{\delta P_{G+1}}{\partial V_{5}} & 1 \\
\frac{\partial P_{G+2}}{\partial \delta_{G+1}} & \frac{\partial P_{G+2}}{\partial \delta_{G+2}} & \cdots \frac{\partial P_{G+2}}{\partial \delta_{N}} & & \frac{\partial P_{G+2}}{\partial V_{G+1}} & \frac{\partial P_{G+2}}{\partial V_{G+2}} & \cdots & \frac{\partial P_{G+2}}{\partial V_{N}} & \frac{\partial P_{G+2}}{\partial V_{2}} & \frac{\partial P_{G+2}}{\partial V_{5}} & 2 \\
\vdots & & & & & & & & & & \\
\frac{\partial P_{N}}{\partial \delta_{G+1}} & \frac{\partial P_{N}}{\partial \delta_{G+2}} & \cdots \frac{\partial P_{N}}{\partial \delta_{N}} & & \frac{\partial P_{N}}{\partial V_{G+1}} & \frac{\partial P_{N}}{\partial V_{G+2}} & \cdots & \frac{\partial P_{N}}{\partial V_{N}} & \frac{\partial P_{N}}{\partial V_{2}} & \frac{\partial P_{N}}{\partial V_{5}} & N-G \\
\hline
\frac{\partial Q_{G+1}}{\partial \delta_{G+1}} & \frac{\partial Q_{G+1}}{\partial \delta_{G+2}} & \cdots \frac{\partial Q_{G+1}}{\partial \delta_{N}} & & \frac{\partial Q_{G+1}}{\partial V_{G+1}} & \frac{\partial Q_{G+1}}{\partial V_{G+2}} & \cdots & \frac{\partial Q_{G+1}}{\partial V_{N}} & \frac{\partial Q_{G+1}}{\partial V_{2}} & \frac{\partial Q_{G+1}}{\partial V_{5}} & N-G+1 \\
\frac{\partial Q_{G+2}}{\partial \delta_{G+1}} & \frac{\partial Q_{G+2}}{\partial \delta_{G+2}} & \cdots \frac{\partial Q_{G+2}}{\partial \delta_{N}} & & \frac{\partial Q_{G+2}}{\partial V_{G+1}} & \frac{\partial Q_{G+2}}{\partial V_{G+2}} & \cdots & \frac{\partial Q_{G+2}}{\partial V_{N}} & \frac{\partial Q_{G+2}}{\partial V_{N}} & \frac{\partial Q_{G+2}}{\partial V_{5}} & \\
\vdots & & & & & & & & & & \\
\frac{\partial Q_{N}}{\partial \delta_{G+1}} & \frac{\partial Q_{N}}{\partial \delta_{G+2}} & \cdots \frac{\partial Q_{N}}{\partial \delta_{N}} & & \frac{\partial Q_{N}}{\partial V_{G+1}} & \frac{\partial Q_{N}}{\partial V_{G+2}} & \cdots & \frac{\partial Q_{N}}{\partial V_{N}} & \frac{\partial Q_{N}}{\partial V_{2}} & \frac{\partial Q_{N}}{\partial V_{5}} & K=2(N-G) \\
\hline
\frac{\partial Q_{2}}{\partial \delta_{G+1}} & \frac{\partial Q_{2}}{\partial \delta_{G+2}} & \cdots \frac{\partial Q_{2}}{\partial \delta_{N}} & & \frac{\partial Q_{2}}{\partial V_{G+1}} & \frac{\partial Q_{2}}{\partial V_{G+2}} & \cdots & \frac{\partial Q_{2}}{\partial V_{N}} & \frac{\partial Q_{2}}{\partial V_{2}} & \frac{\partial Q_{2}}{\partial V_{5}} & 2(N-G+1) \\
\frac{\partial Q_{5}}{\partial \delta_{G+1}} & \frac{\partial Q_{5}}{\partial \delta_{G+2}} & \cdots \frac{\partial Q_{5}}{\partial \delta_{N}} & & \frac{\partial Q_{5}}{\partial V_{G+1}} & \frac{\partial Q_{5}}{\partial V_{G+2}} & \cdots & \frac{\partial Q_{5}}{\partial V_{N}} & \frac{\partial Q_{5}}{\partial V_{2}} & \frac{\partial Q_{5}}{\partial V_{5}} & K' \\
\hline
\multicolumn{1}{c}{1} & 2 & \cdots\ N-G & & N-G+1 & & \cdots\cdots & k=2(N-G) & K+1 & \multicolumn{1}{c}{K'} &
\end{array}
$$

Compute the $K' \times j'$ matrix $[B']$ as follows.

for all the generator buses and thus from the vectors $[Q_{min}]$ and $[Q_{max}]$ compare $[Q]$ with $[Q_{min}]$ and $[Q_{max}]$ and check for limit violation. Temporarily include all the Q_i's which violate the limits in the vector [a], forming the augmented vector [a']. Include the violated limits themselves in the vector [s] forming he augmented vector [s']. Accordingly regroup the state variables [x] and [y] as [x'] and [y'] . For example, suppose it is found that $Q_2 < Q_{2\ min}$ and $Q_5 > Q_{5\ max}$ ie two of the elements in [Q] have limit violation. Then temporarily define the various vectors and indices as follows.

$$K' = K + 2 \ ; \qquad j' = j - 2$$

$$\underset{K' \times 1}{[\,x'\,]} = \begin{bmatrix} \delta_{G+1} \\ \delta_{G+2} \\ \vdots \\ \delta_N \\ \hdashline V_{G+1} \\ V_{G+2} \\ \vdots \\ VN \\ \cdots\cdots \\ V_2 \\ V_5 \end{bmatrix} \qquad \underset{j' \times 1}{[\,y'\,]} = \begin{bmatrix} \delta_2 \\ \delta_3 \\ \vdots \\ \delta_G \\ V_1 \\ V_3 \\ V_4 \\ V_6 \\ V_7 \\ \vdots \\ V_G \end{bmatrix}$$

$$[B'] = \left[\begin{array}{ccccccccccc|l}
\frac{\partial P_{G+1}}{\partial \delta_2} & \frac{\partial P_{G+1}}{\partial \delta_3} & \cdots & \frac{\partial P_{G+1}}{\partial \delta_G} & \frac{\partial P_{G+1}}{\partial V_1} & \frac{\partial P_{G+1}}{\partial V_3} & \frac{\partial P_{G+1}}{\partial V_4} & \frac{\partial P_{G+1}}{\partial V_6} & \frac{\partial P_{G+1}}{\partial V_7} & \cdots & \frac{\partial P_{G+1}}{\partial V_G} & 1 \\
\frac{\partial P_{G+2}}{\partial \delta_2} & \frac{\partial P_{G+2}}{\partial \delta_3} & \cdots & \frac{\partial P_{G+2}}{\partial \delta_G} & \frac{\partial P_{G+2}}{\partial V_1} & \frac{\partial P_{G+2}}{\partial V_3} & \frac{\partial P_{G+2}}{\partial V_4} & \frac{\partial P_{G+2}}{\partial V_6} & \frac{\partial P_{G+2}}{\partial V_7} & \cdots & \frac{\partial P_{g+2}}{\partial V_G} & 2 \\
\vdots & & & & & & & & & & & \\
\frac{\partial P_N}{\partial \delta_2} & \frac{\partial P_N}{\partial \delta_3} & \cdots & \frac{\partial P_N}{\partial \delta_G} & \frac{\partial P_N}{\partial V_1} & \frac{\partial P_N}{\partial V_3} & \frac{\partial P_N}{\partial V_4} & \frac{\partial P_N}{\partial V_6} & \frac{\partial P_N}{\partial V_7} & \cdots & \frac{\partial P_N}{\partial V_G} & N-G \\
\hdashline
\frac{\partial Q_{G+1}}{\partial \delta_2} & \frac{\partial Q_{G+1}}{\partial \delta_3} & \cdots & \frac{\partial Q_{G+1}}{\partial \delta_G} & \frac{\partial Q_{G+1}}{\partial V_1} & \frac{\partial Q_{G+1}}{\partial V_3} & \frac{\partial Q_{G+1}}{\partial V_4} & \frac{\partial Q_{G+1}}{\partial V_6} & \frac{\partial Q_{G+1}}{\partial V_7} & \cdots & \frac{\partial Q_{G+1}}{\partial V_G} & N-G+1 \\
\frac{\partial Q_{G+2}}{\partial \delta_2} & \frac{\partial Q_{G+2}}{\partial \delta_3} & \cdots & \frac{\partial Q_{G+2}}{\partial \delta_G} & \frac{\partial Q_{G+2}}{\partial V_1} & \frac{\partial Q_{G+2}}{\partial V_3} & \frac{\partial Q_{G+2}}{\partial V_4} & \frac{\partial Q_{G+2}}{\partial V_6} & \frac{\partial Q_{G+2}}{\partial V_7} & \cdots & \frac{\partial Q_{G+2}}{\partial V_G} & N-G+2 \\
\vdots & & & & & & & & & & & \\
\frac{\partial Q_N}{\partial \delta_2} & \frac{\partial Q_N}{\partial \delta_3} & \cdots & \frac{\partial Q_N}{\partial \delta_G} & \frac{\partial Q_N}{\partial V_1} & \frac{\partial Q_N}{\partial V_3} & \frac{\partial Q_N}{\partial V_4} & \frac{\partial Q_N}{\partial V_6} & \frac{\partial Q_N}{\partial V_7} & \cdots & \frac{\partial Q_N}{\partial V_G} & 2(N-G) \\
\frac{\partial Q_2}{\partial \delta_2} & \frac{\partial Q_2}{\partial \delta_3} & \cdots & \frac{\partial Q_2}{\partial \delta_G} & \frac{\partial Q_2}{\partial V_1} & \frac{\partial Q_2}{\partial V_3} & \frac{\partial Q_2}{\partial V_4} & \frac{\partial Q_2}{\partial V_6} & \frac{\partial Q_2}{\partial V_7} & \cdots & \frac{\partial Q_2}{\partial V_G} & 2(N-G)+1 \\
\frac{\partial Q_5}{\partial \delta_2} & \frac{\partial Q_5}{\partial \delta_3} & \cdots & \frac{\partial Q_5}{\partial \delta_G} & \frac{\partial Q_5}{\partial V_1} & \frac{\partial Q_5}{\partial V_3} & \frac{\partial Q_5}{\partial V_4} & \frac{\partial Q_5}{\partial V_6} & \frac{\partial Q_5}{\partial V_7} & \cdots & \frac{\partial Q_5}{\partial V_G} & K' \\
\hline
1 & 2 & & G-1 & G & G+1 & & & & & &
\end{array}\right]$$

Compute the m x K' matrix [C'] as follows:

$$[c'] = \begin{array}{cccccccccc|c}
\frac{\partial P_1}{\partial \delta_{G+1}} & \frac{\partial P_1}{\partial \delta_{G+2}} & \cdots & \frac{\partial P_1}{\partial \delta_N} & \frac{\partial P_1}{\partial V_{G+1}} & \frac{\partial P_1}{\partial V_{G+2}} & \cdots & \frac{\partial P_1}{\partial V_N} & \frac{\partial P_1}{\partial V_2} & \frac{\partial P_1}{\partial V_5} & 1 \\
\frac{\partial P_2}{\partial \delta_{G+1}} & \frac{\partial P_2}{\partial \delta_{G+2}} & \cdots & \frac{\partial P_2}{\partial \delta_N} & \frac{\partial P_2}{\partial V_{G+1}} & \frac{\partial P_2}{\partial V_{G+2}} & \cdots & \frac{\partial P_2}{\partial V_N} & \frac{\partial P_2}{\partial V_2} & \frac{\partial P_2}{\partial V_5} & 2 \\
\vdots & & & & & & & & & & \\
\frac{\partial P_G}{\partial \delta_{G+1}} & \frac{\partial P_G}{\partial \delta_{G+2}} & \cdots & \frac{\partial P_G}{\partial \delta_N} & \frac{\partial P_G}{\partial V_{G+1}} & \frac{\partial P_G}{\partial V_{G+2}} & \cdots & \frac{\partial P_G}{\partial V_N} & \frac{\partial P_G}{\partial V_2} & \frac{\partial P_G}{\partial V_5} & G \\
\hline
0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & G+1 \\
0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 1 & 0 & G+2 \\
0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \\
0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \\
0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & 1 & \\
0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \\
\vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & \\
0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 2G \\
\hline
0 & 0 & \cdots & 0 & 1 & 0 & \cdots & 0 & 0 & 0 & 2G+1 \\
0 & 0 & \cdots & 0 & 0 & 1 & \cdots & 0 & 0 & 0 & \\
\vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & \\
0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 1 & 0 & 0 & m=N+G \\
\hline
1 & 2 & & N-G & N-G+1 & & & K=2(N-G) & K+1 & K' &
\end{array}$$

Compute the m x j' matrix [D'] as follows:

$$[D'] = \left[\begin{array}{ccccccccccc|c}
\frac{\partial P_1}{\partial \delta_2} & \frac{\partial P_1}{\partial \delta_3} & \cdots & \frac{\partial P_1}{\partial \delta_G} & \frac{\partial P_1}{\partial V_1} & \frac{\partial P_1}{\partial V_3} & \frac{\partial P_1}{\partial V_4} & \frac{\partial P_1}{\partial V_6} & \frac{\partial P_1}{\partial V_7} & \cdots & \frac{\partial P_1}{\partial V_G} & 1 \\
\frac{\partial P_2}{\partial \delta_2} & \frac{\partial P_2}{\partial \delta_3} & \cdots & \frac{\partial P_2}{\partial \delta_G} & \frac{\partial P_2}{\partial V_1} & \frac{\partial P_2}{\partial V_3} & \frac{\partial P_2}{\partial V_4} & \frac{\partial P_2}{\partial V_6} & \frac{\partial P_2}{\partial V_7} & \cdots & \frac{\partial P_2}{\partial V_G} & 2 \\
\vdots & & & & & & & & & & & \\
\frac{\partial P_G}{\partial \delta_2} & \frac{\partial P_G}{\partial \delta_3} & \cdots & \frac{\partial P_G}{\partial \delta_G} & \frac{\partial P_G}{\partial V_1} & \frac{\partial P_G}{\partial V_3} & \frac{\partial P_G}{\partial V_4} & \frac{\partial P_G}{\partial V_6} & \frac{\partial P_G}{\partial V_7} & \cdots & \frac{\partial P_G}{\partial V_G} & G \\
\hline
0 & 0 & \cdots & 0 & 1 & 0 & 0 & 0 & 0 & \cdots & 0 & G+1 \\
0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & G+2 \\
0 & 0 & \cdots & 0 & 0 & 1 & 0 & 0 & 0 & \cdots & 0 & \\
0 & 0 & \cdots & 0 & 0 & 0 & 1 & 0 & 0 & \cdots & 0 & \\
0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & \\
0 & 0 & \cdots & 0 & 0 & 0 & 0 & 1 & 0 & \cdots & 0 & \\
0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 1 & \cdots & 0 & \\
\vdots & & & & & & & & & & & \\
0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 2G \\
\hline
0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 2G+1 \\
0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & \\
\vdots & & & & & & & & & & & \\
0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & m=N+G \\
\hline
1 & 2 & \cdots & G-1 & G & G+1 & \cdots & & & & j &
\end{array}\right]$$

Step 5

compute $[A']^{-1}$ and store it in the same location as $[A']$ and compute $[C'] [A']^{-1}$ and store it in the same location as $[C']$

Compute $[G']$ $[D'] - [C'] [A']^{-1} [B]$ and store it in the same location as $[D']$

Compute $[A']^{-1} [B']$ and store it in the same location as $[B']$

Compute $[H] = [G']^{tr} [W]$ where $[H]$ is a J' x m matrix

Compute $[H] [G']$ and store it in the upper part of the same location as $[G']$

Compute $([H] [G'])^{-1}$ and store it in the same location as $[H] [G']$

Compute $[A']^{-1} ([S'] - [a']$ and store it in the same location as $[S'] - [a']$

Compute $[t] - [b] -]c'] [A']^{-1} ([s'] - [a]$ and store it in the same location as $[t] - [b]$

Compute $[\Delta y'] = ([H] [G'])^{-1} [H] ([t] - [b] - [c'] [A']^{-1} ([s'] - [a']))$

from equation 7

Compute $[\Delta x'] = [A']^{-1} ([s'] - [a'] - [A']^{-1} [B'] [\Delta y']$

From equation 6

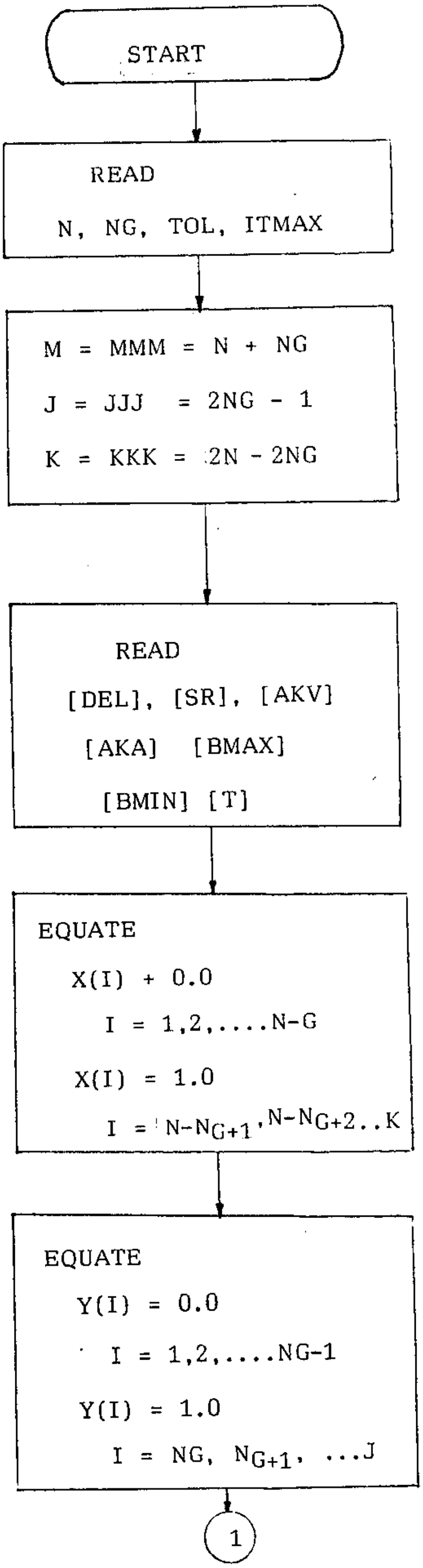
Step 6

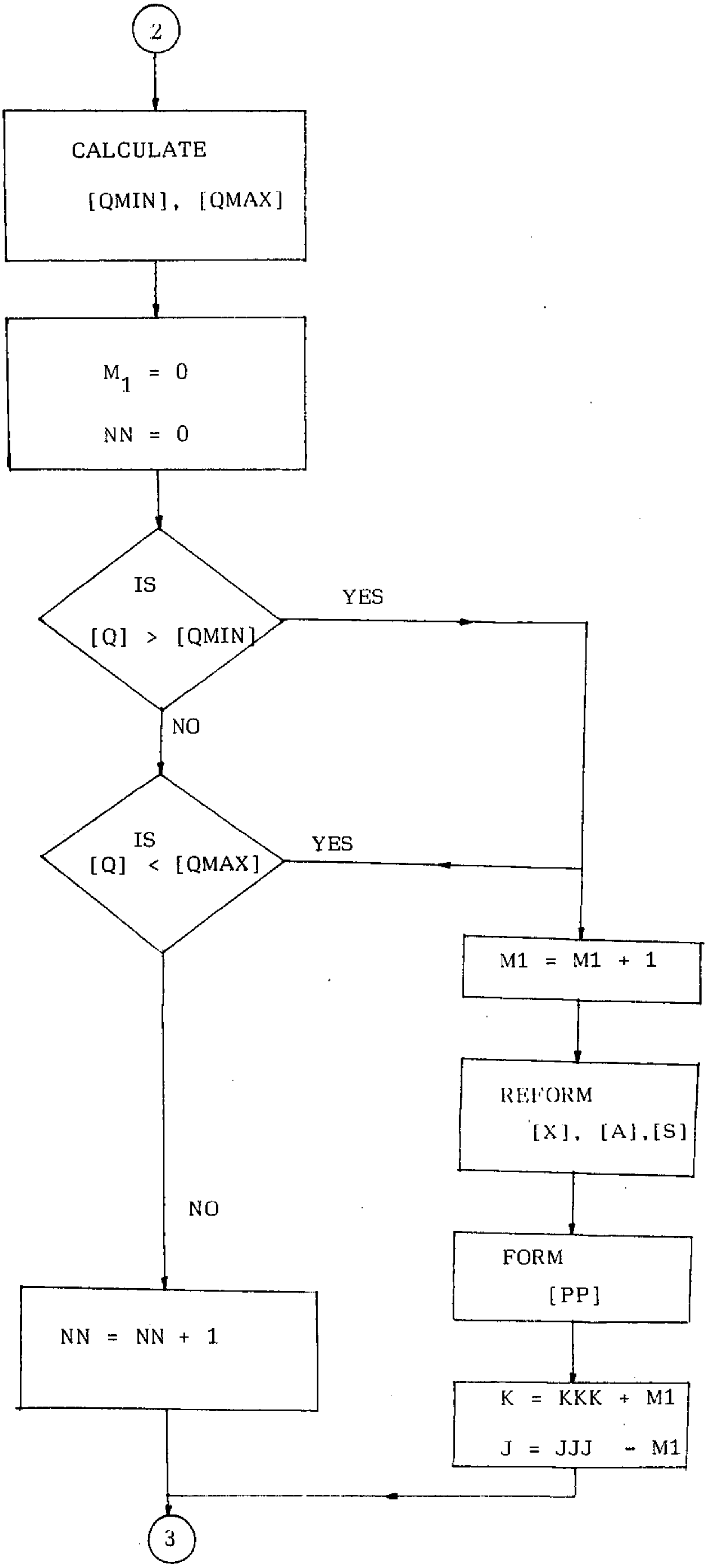
Check whether all the elements is $[\Delta x']$ and $[\Delta y']$ are less than a pre specified ε in magnitude of so, the solution has converged. If not go to step 7.

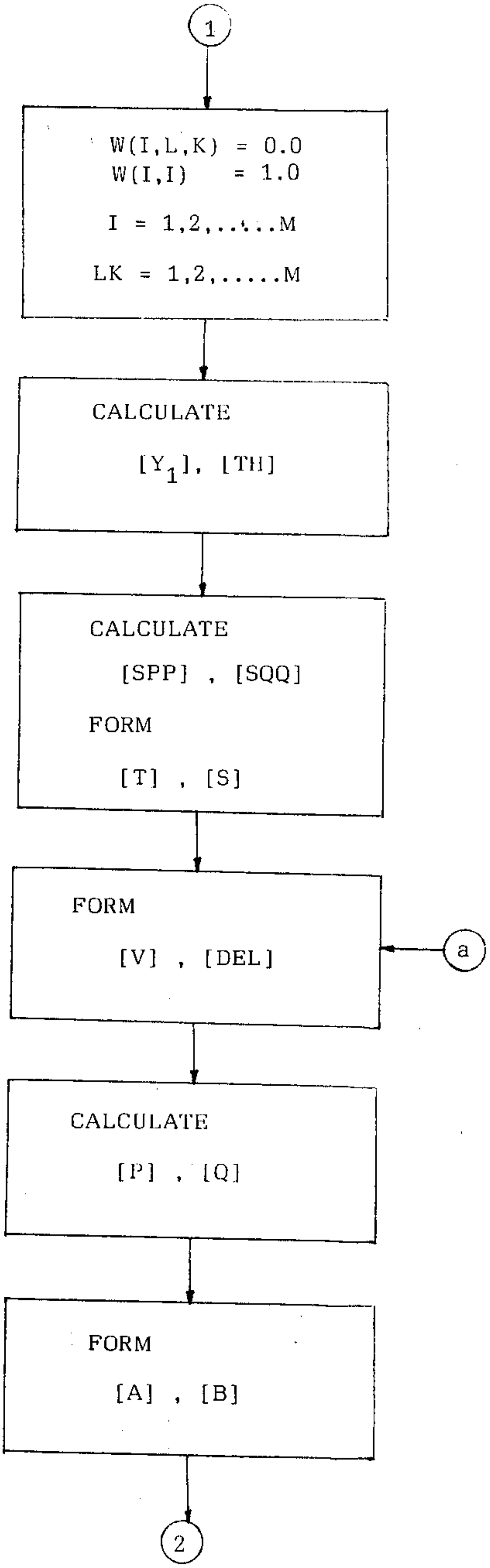
flow diagram

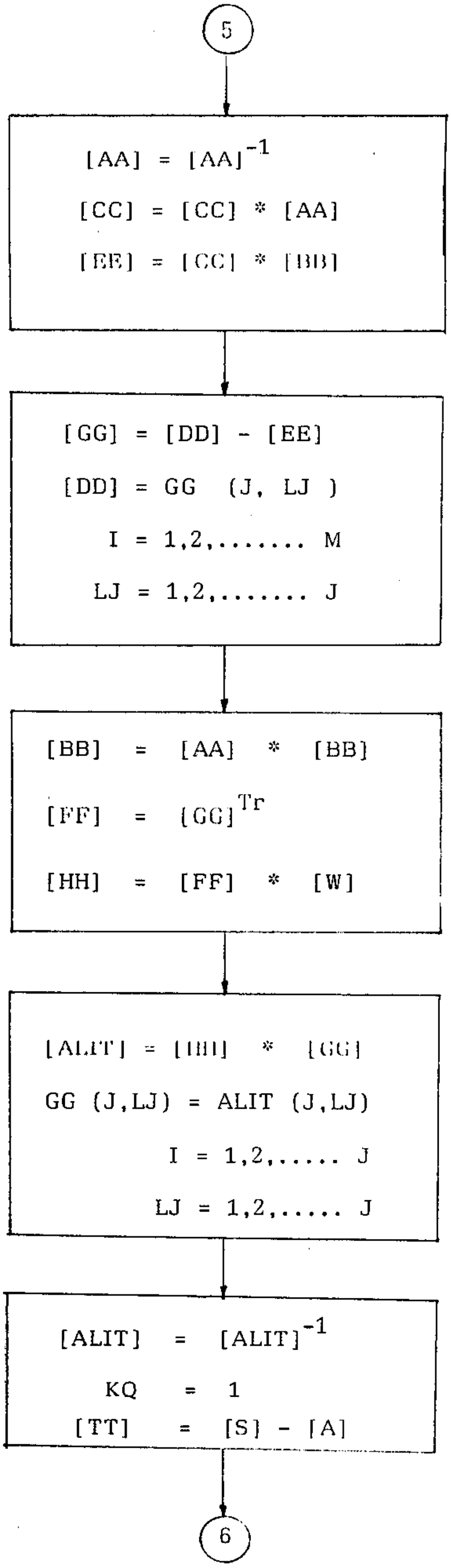
CHAPTER VI

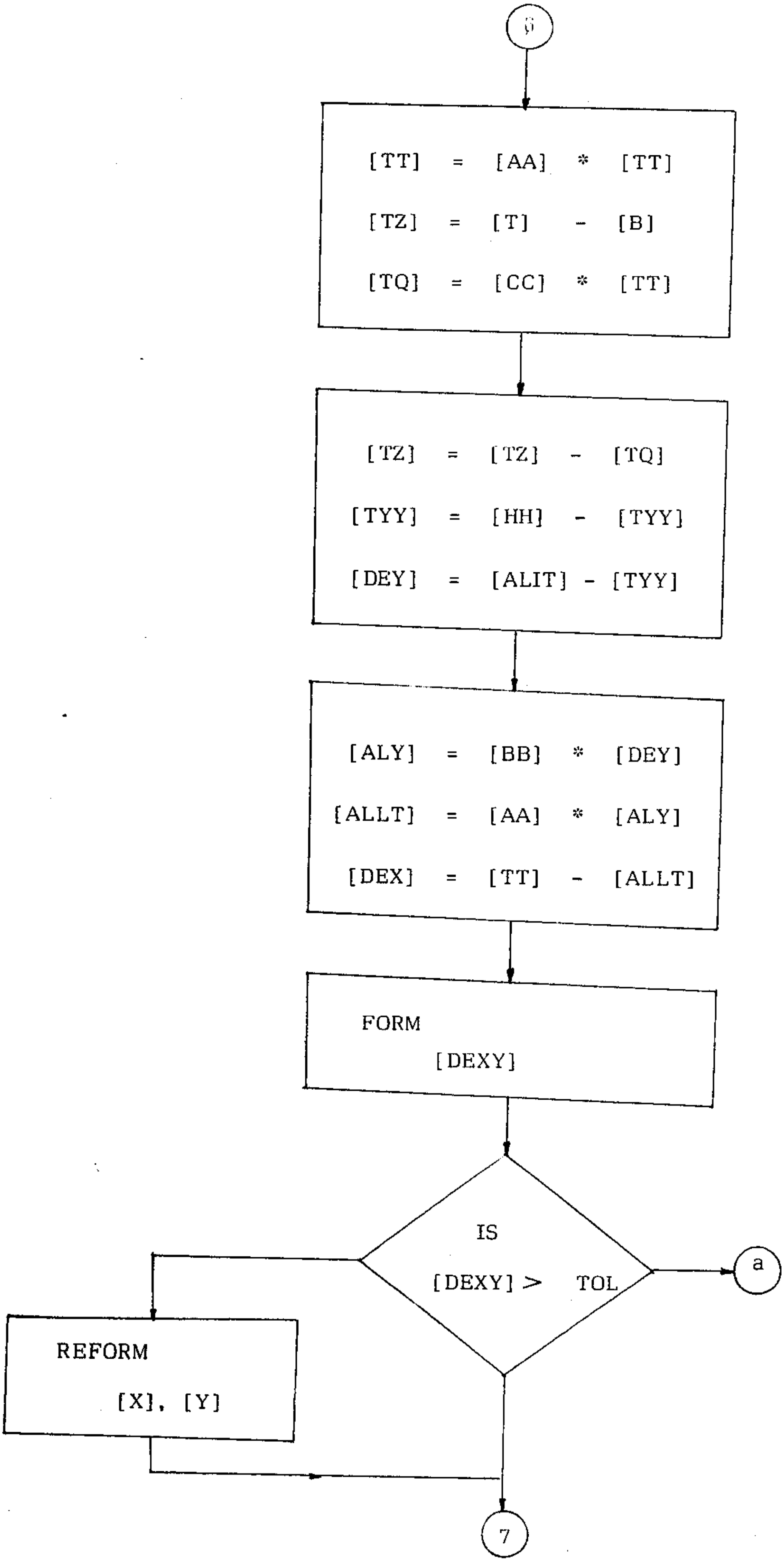
FLOW DIAGRAM:

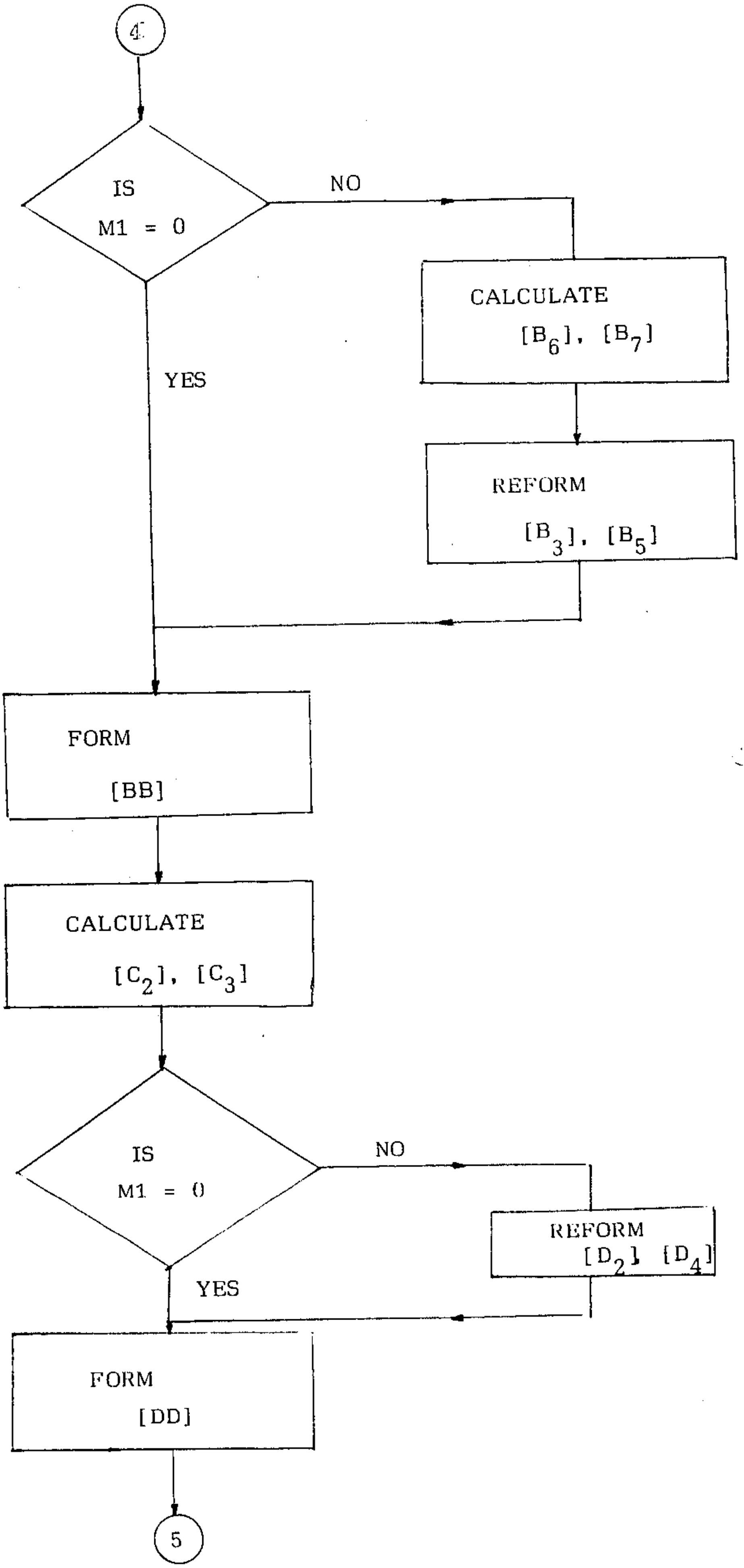


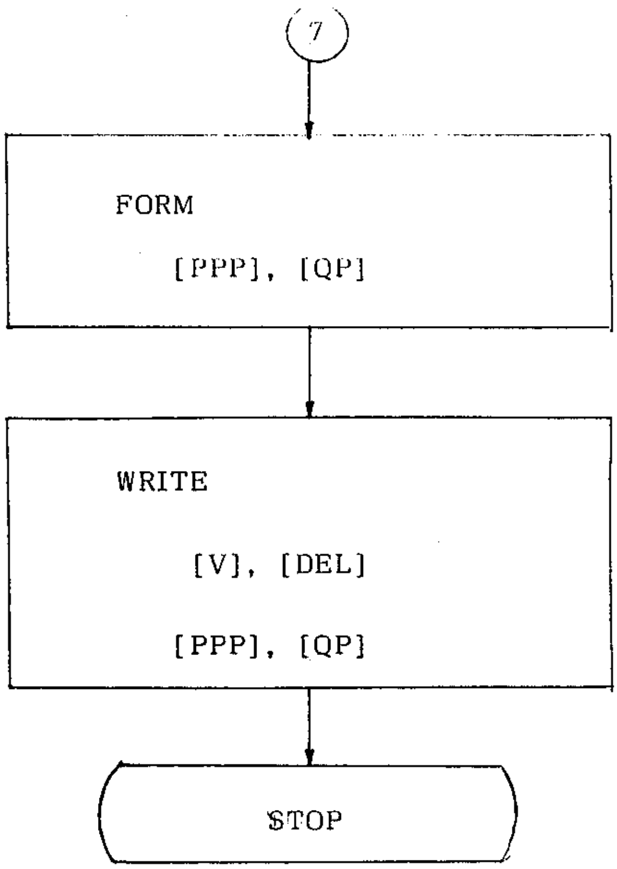

2
CALCULATE
[QMIN], [QMAX]
M1 = 0
NN = 0
IS
[Q] > [QMIN]
YES
NO
IS
[Q] < [QMAX]
YES
M1 = M1 + 1
REFORM
[X], [A],[S]
NO
FORM
[PP]
NN = NN + 1
K = KKK + M1
J = JJJ - M1
3


1
W(I,L,K) = 0.0
W(I,I) = 1.0
I = 1,2,......M
LK = 1,2,.....M
CALCULATE
$[Y_1]$, [TH]
CALCULATE
[SPP] , [SQQ]
FORM
[T] , [S]
FORM
[V] , [DEL]
a
CALCULATE
[P] , [Q]
FORM
[A] , [B]
2


5
[AA] = [AA]-1
[CC] = [CC] * [AA]
[EE] = [CC] * [BB]
[GG] = [DD] - [EE]
[DD] = GG (J, LJ)
I = 1,2,....... M
LJ = 1,2,....... J
[BB] = [AA] * [BB]
[FF] = [GG]Tr
[HH] = [FF] * [W]
[ALIT] = [HH] * [GG]
GG (J,LJ) = ALIT (J,LJ)
I = 1,2,..... J
LJ = 1,2,..... J
[ALIT] = [ALIT]-1
KQ = 1
[TT] = [S] - [A]
6


6
[TT] = [AA] * [TT]
[TZ] = [T] - [B]
[TQ] = [CC] * [TT]
[TZ] = [TZ] - [TQ]
[TYY] = [HH] - [TYY]
[DEY] = [ALIT] - [TYY]
[ALY] = [BB] * [DEY]
[ALLT] = [AA] * [ALY]
[DEX] = [TT] - [ALLT]
FORM
[DEXY]
IS
[DEXY] > TOL
a
REFORM
[X], [Y]
7


4
IS
M1 = 0
NO
YES
CALCULATE
[B_6], [B_7]
REFORM
[B_3], [B_5]
FORM
[BB]
CALCULATE
[C_2], [C_3]
IS
M1 = 0
NO
YES
REFORM
[D_2], [D_4]
FORM
[DD]
5


7
FORM
[PPP], [QP]
WRITE
[V], [DEL]
[PPP], [QP]
STOP

software and results

```
      DIMENSION DEL(5),SR(2),AKV(5),AKA(5),BMAX(7),BMIN(7),C4(10,10)
      DIMENSION CZ(10,10),S1(8,8),QP(5),PPP(5),ALLT(8,8),A6(3,2)
      DIMENSION CH(10,10),F(8,8),X(5),Y(5),U(8,8),SPP(5),S(8,8)
      DIMENSION A2(3,3),A8(2,3),A7(3,2),A9(2,3),A10(2,2),T7(8,8)
      DIMENSION AA(8,8),B2(3,1),B4(3,1),B3(3,2),B5(3,2),B6(2,2)
      DIMENSION B7(2,2),B8(8,8),C2(2,3),C5(2,3),C6(2,2),C7(2,2)
      DIMENSION C9(3,3),CC(8,8),D1(2,1),D2(2,2),D4(2,2),DEXY(8,8)
      DIMENSION EE(8,8),GG(8,8),FF(8,8),HH(8,8),ALIT(8,8),TT(8,8)
      DIMENSION TQ(8,8),TYY(8,8),DEY(8,8),DEX(8,8),Y1(10,10)
      DIMENSION ALY(8,8),BQQ(5),PP(2),A3(3,3),A4(3,3),A5(3,3)
      DIMENSION IDY(5),V(5),A(8,8),P(5),B(8,8),QMIN(2),QMAX(2),Q(5)
      DIMENSION B1(8,8),TH(10,10),BD(8,8),CA(10,10),CY(10,10)
      COMPLEX CH,CZ,CY,CA
      OPEN(2,FILE='S.DAT'=,STATUS='OLD')

C     ####################################################
C     #################      MAIN     PROGRAM      ###################
C     ####################################################
      READ(2,199) N,NG,TOL,ITMS
      M=N+NG
      J=2*NG-1
      K=2*(N-NG)
      MMM=M
      JJJ=J
```

```
      KKK=K
      READ(2,200) DEL(I)
      READ(2,201) (BR(I),I=1,NG)
      READ(2,202) (AKV(I),I=1,N)
      READ(2,202) (AKA(I),I=1,N)
      READ(2,203) (BMAX(I),I=1,M)
      READ(2,203) (BMIN(I),I=1,M)
      DO 1 I=1,N
      READ(2,204) (CZ(I,LJ),LJ=1,N)
1     CONTINUE
      DO 2 I=1,N
      READ(2,204) (CM(I,LJ),LJ=1,N)
2     CONTINUE
      READ(2,205) (T(I,1),I=NG+1,M)
C     ##############################################################
C     ######   [X]  :   DEPENDENT  VARIABLES   #####################
C     ##############################################################
      DO 3 I=1,KKK
      IF(I.GE.(N-NG+1)) GOTO 4
      X(I)=0.0
      GOTO 3
4     X(I)=1.0
3     CONTINUE
C     ##############################################################
C     ##########   [Y]  :   INDEPENDENT VARIABLES ##################
C     ##############################################################
      DO 5 I=1,JJJ
      IF(I.GE.NG) GOTO 6
      Y(I)=0.0
```

```
      GOTO 5
 6    Y(I)=1.0
 5    CONTINUE
C     #############################################################
C     #############     [W]   :    WEIGHTAGES            ################
C     #############################################################
      DO 7 I=1,M
      DO 8 LK=1,M
      IF(I.EQ.LK) GOTO 9
      W(I,LK)=0.0
      GOTO 8
 9    W(I,LK)=1.0
 8    CONTINUE
 7    CONTINUE
      CALL YBUS(CZ,CH,Y1,TH,N)
      CALL NR (AKV,AKA,Y1,TH,N,SPP,SQQ)
C     ##############################################################
C     ##############     [T]   :    TARGET VALUE        ##############
C     ##############################################################
      DO 10 I=1,NG
      T(I,1)=SPP(I)
 10   CONTINUE
C     ##############################################################
C     ##########       [S]   :    SPECIFIED VALUES       #################
C     ##############################################################
      DO 11 I=1,KKK
      IF(I.GE.(N-NG+1)) GOTO 12
      S(I,1)=SPP(NG+I)
      GOTO 11
 12
```

```
C      ################################################################
C      ##  LOAD FLOW AS A PLANNING TOOL - AN OPTIMAL APPROACH        ##
C      ################################################################


       WRITE(*,*) '     LOAD FLOW AS A PLANNING TOOL                       '
       WRITE(*,*) '                                 - AN OPTIMAL APPROACH '
       WRITE(*,*) '                                                        '
       WRITE(*,*) '                                                        '
       WRITE(*,*) '              WORK DONE BY:                             '
       WRITE(*,*) '                             R.SHANTHI                  '
       WRITE(*,*) '                             M.JAYA KUMAR               '
       WRITE(*,*) '                             V.S.RAVICHANDRAN           '
       WRITE(*,*) '                             G.SHANTHANAYAKI            '
       WRITE(*,*) '                             SHINY PURUSHOTHAMAN        '
       WRITE(*,*) '                                                        '
       WRITE(*,*) '                                                        '
       WRITE(*,*) '              GUIDED BY:                                '
       WRITE(*,*) '                             Mr.K.SURENDRAN M.Tech.,    '
```

```
      S(I,1)=SQQ(I-N+2*NG)
   11 CONTINUE
C     ##############################################################
C     ####################   ITERATION STARTS   ######################
C     ##############################################################
      DO 2000 IAB=1,ITMAX
      DO 13 I=1,JJJ
   13 IDY(I)=1
C     ##############################################################
C     ##############     [V]   :    VOLTAGE  MAGNITUDE       ############
C     ##############     [DEL]:    VOLTAGE  ANGLE  IN  RADIANS ############
C     ##############################################################
      DO 14 I=1,N
      IF(I.GT.NG) GOTO 15
      V(I)=Y(I+NG-1)
      IF(I.EQ.1) GOTO 16
      DEL(I)=Y(I-1)
   16 GOTO 14
   15 V(I)=X(I+N-2*NG)
      DEL(I)=X(I-NG)
   14 CONTINUE
      CALL NR(V,DEL,Y1,TH,N,P,Q)
C     ##############################################################
C     ##############    [A]   :    SPECIFIED  VARIABLES    ################
C     ##############################################################
      DO 17 I=1,N-NG
   17 A(I,1)=P(NG+I)
      DO 18 I=N-NG+1,KKK
   18 A(I,1)=Q(2*NG+I-N)
C
```

```
        ################################################
        ########### [B]  :  TARGET VARIABLES  ##################
C       ################################################
C       ################################################

        DO 19 I=1,NG

  19    B(1,1)=P()

        DO 20 I=NG+1,MMM

  20    B(I,1)=V(I-NG)

        ################################################
C       #####  [QMAX]   :   MAXIMUM VALUE OF REACTIVE POWER   ######
C       #####  [QMIN]   :   MINIMUM VALUE OF REACTIVE POWER ######
C       ################################################
C       ################################################

        DO 21 I=1,NG

        QMIN(I)=-SQRT(SR(I)*SR(I)-P(1)*P(I))

  21    QMAX(I)=ABS(QMIN(I))

C       ################################################
C       ##########  CHECK FOR REACTIVE POWER VIOLATION  ############
C       ################################################

        M1=0

        NN=0

        DO 22 I=1,NG

        IF(Q(I).GT.QMIN(I)) GOTO 23

        M1=M1+1

        X(KKK+M1)=V(I)

        A(KKK+M1,1)=Q(I)

        IDY(I+NG-1)=0

        PP(I-NN)=I

        S(KKK+M1,1)=QMIN(I)

        K=KKK+M1

        J=JJJ-M1

        GOTO 22

  23    IF(Q(I).LT.QMAX(I)) GOTO 24

        M1=M1+1
```

```
33      CONTINUE
        DO 35 I=1,M1
        DO 36 KJ=1,M1
        JI1=PP(I)
        JI2=PP(KJ)
        IF(JI1.EQ.JI2) GOTO 37
        THY1 = TH(JI1,JI2)+DEL(JI1)-DEL(JI2)
        A10(I,KJ)=ABS(V(JI1)*Y1(JI1,JI2))*SIN(THY1)
        GOTO 36
37      A10(I,KJ)=0
        DO 38 LK=1,N
        IF(LK.EQ.JI1) GOTO 38
        THIY = TH(JI1,LK)+DEL(JI1)-DEL(LK)
        A10(I,KJ)=A10(I,KJ)+ABS(V(LK)*Y1(JI1,LK))*SIN( THIY )
38      CONTINUE
        A10(I,KJ)=A10(I,KJ)+2*ABS(V(JI1)*Y1(JI1,JI1))*SIN(TH(JI1,JI1))
36      CONTINUE
35      CONTINUE
C       #############################################################
C       ################    FORMATION OF [AA] MATRIX    #############
C       #############################################################
32      DO 40 I=1,N-NG
        DO 40 LJ=1,N-NG
40      AA(I,LJ)=A2(I,LJ)
        DO 41 I=1,N-NG
        DO 41 LJ=N-NG+1,KKK
41      AA(I,LJ)=A3(I,LJ-N+NG)
        DO 42 I=N-NG+1,KKK
        DO 42 LJ=1,N-NG
```

```
      A5(IT,IL)=ABS(V(I)*Y1(I,LJ))*SIN(TH(I,LJ)+DEL(I)-DEL(LJ))
      GOTO 29
30    A2(IT,IL)=0.0
      A3(IT,IL)=0.0
      A4(IT,IT)=0.0
      A5(IT,IL)=0.0
      DO 31 LL=1,N
      IF(LL.EQ.I) GOTO 31
      THT = TH(I,LL)+DEL(I)-DEL(LL)
      A2(IT,IL)=-ABS(V(I)*V(LL)*Y1(I,LL))*SIN(THT)+A2(IT,IL)
      A3(IT,IL)=ABS( V(LL)*Y1(I,LL) ) * COS( THT )+ A3(IT,IL)
      A4(IT,IL)=ABS(V(I)*V(LL)*Y1(I,LL))*COS(THT )+ A4(IT,IL)
      A5(IT,IL)=ABS(V(LL)*Y1(I,LL))*SIN( THT )+A5(IT,IL)
31    CONTINUE
      A3(IT,IL)=2*ABS(V(I)*Y1(I,I))*COS(TH(I,I))+A3(IT,IL)
      A5(IT,IL)=2*ABS(V(I)*Y1(I,I))*SIN(TH(I,I))+A5(IT,IL)
29    CONTINUE
28    CONTINUE
      IF(M1.EQ.0) GOTO 32
      DO 33 I=1,M1
      NN1=PP(I)
      DO 34 K1=NG+1,N
      THH = TH(K1,NN1)+DEL(K1)-DEL(NN1)
      THY = TH(NN1,K1)+DEL(NN1)-DEL(K1)
      A6(K1-NG,I)=ABS(V(K1)*Y1(K1,NN1))*COS(THH )
      A8(I,K1-NG)=-ABS(V(K1)*V(NN1)*Y1(NN1,K1))*COS(THY )
      A7(K1-NG,I)=ABS(V(K1)*Y1(K1,NN1))*SIN(THH)
      A9(I,K1-NG)=ABS(V(NN1)*Y1(NN1,K1))*SIN(THY)
34    CONTINUE
```

```
       X(KKK+M1)=V(I)

       A(KKK+M1,1)=Q(I)

       IDY(NG-1+I)=0

       PP(I-NN)=I

       S(KKK+M1,1)=QMAX(I)

       K=KKK+M1

       J=JJJ-M1

       GOTO 22

  24   NN=NN+1

  22   CONTINUE

C      ######################################################################
C      #############     CHECK FOR TARGET VIOLATION    ###############
C      ######################################################################

       DO 25 I=1,MMM

       IF(B(I,1).GT.BMIN(I).AND.B(I,1).LT.BMAX(I)) GOTO 25

       W(I,I)=W(I,I)+1.0

  25   CONTINUE

C      ############################################################
C      #FORMATION OF [A2],[A3],[A4],[A5],[A6],[A7],[A8],[A9],[A10]#
       ############################################################

       DO 28 I=NG+1,N

       DO 29 LJ=NG+1,N

       IT=I-NG

       IL=LJ-NG

       IF(I.EQ.LJ) GOTO 30

       THIP=TH(I,LJ)+DEL(I)-DEL(LJ)

       A2(IT,IL)=ABS(V(I)*V(LJ)*Y1(I,LJ))*SIN(THIP)

       A3(IT,IL)=ABS(V(I)*Y1(I,LJ))*COS(THIP)

       THT1=TH(I,LJ)+DEL(I)-DEL(LJ)

       A4(IT,IL)=- ABS( V(I)*V(LJ)*Y1(I,LJ) )*COS( THT1)
```

```
 42       AA(I,LJ)=A4(I-N+NG,LJ)
          DO 43 I=N-NG+1,KKK
          DO 43 LJ=N-NG+1,KKK
 43       AA(I,LJ)=A5(I-N+NG,LJ-N+NG)
          IF(M1.EQ.0) GO TO 44
          DO 45 I=1,N-NG
          DO 45 LJ=KKK+1,K
 45       AA(I,LJ)=A6(I,LJ-KKK)
          DO 46 I=N-NG+1,KKK
          DO 46 LJ=KKK+1,K
 46       AA(I,LJ)=A7(I-N+NG,LJ-KKK)
          DO 47 I=KKK+1,K
          DO 47 LJ=1,N-NG
 47       AA(I,LJ)=A8(I-KKK,LJ)
          DO 48 I=KKK+1,K
          DO 48 LJ=N-NG+1,KKK
 48       AA(I,LJ)=A9(I-KKK,LJ-N+NG)
          DO 49 I=KKK+1,K
          DO 49 LJ=KKK+1,K
 49       AA(I,LJ)=A10(I-KKK,LJ-K)
C         ########################################################################
C         ###########     FORMATOIN OF [B2],[B3],[B4],[B5],[B6],[B7]     #######
C         ########################################################################
 44       DO 50 I=NG+1,N
          DO 51 LJ=1,NG
          LG1=I-NG
          UHI=TH(I,LJ)+DEL(I)-DEL(LJ)
          IF(LJ.EQ.1) GOTO 52
          B2(LG1,LJ-1)=ABS(V(I)*V(LJ)*Y1(I,LJ))*SIN(UHI)
          B4(LG1,LJ-1)=-ABS(V(I)*V(LJ)*Y1(I,LJ))*COS(UHI)
```

```
52        B3(LG1,LJ)=ABS(V(I)*Y1(I,LJ))*COS(UHI)
          B5(LG1,LJ)=ABS(V(I)*Y1(I,LJ))*SIN(UHI)
51        CONTINUE
50        CONTINUE
          IF(M1.EQ.0) GOTO 53
          DO 54 I=1,M1
          NV=PP(I)
          DO 54 IJK=1,N-NG
          B3(IJK,NV)=B3(IJK,NV+1)
54        B5(IJK,NV)=B5(IJK,NV+1)
          DO 55 II=1,M1
          NA=PP(II)
          MJ=0
          DO 56 LJ=1,NG
          IF(NA.EQ.LJ) GOTO 57
          HIJ=TH(NA,LJ)+DEL(NA)-DEL(LJ)
          B7(I,LJ-MJ)=ABS(V(NA)*Y1(NA,LJ))*SIN(HIJ)
          GOTO 56
57        MJ=MJ+1
56        CONTINUE
55        CONTINUE
          DO 58 IK=1,M1
          NA=PP(I)
          DO 59 JK=2,NG
          IF(NA.EQ.J) GOTO 60
          THY2 = TH(NA,LJ)+DEL(NA)-DEL(LJ)
          B6(I,LJ-1)=-ABS(V(NA)*V(LJ)*Y1(NA,LJ))*COS(THY2 )
          GOTO 57
```

```
60    B6(NA,LJ-1)=0
      DO 610 KK=1,N
      IF(NA.EQ.KK) GOTO 610
      THUY = TH(NA,KK)+DEL(NA)-DEL(KK)
      B6(NA,LJ-1)=ABS(V(NA)*V(KK)*Y1(NA,KK))*COS(THUY) + B6(NA,LJ-1)
610   CONTINUE
59    CONTINUE
58    CONTINUE
C     ########################################################################
C     ###################    FORMATION OF [BB]  MATRIX   ##################
C     ########################################################################
53    DO 600 I = 1,N-NG
      DO 606 JJ = 1,NG-1
      BB(I,JJ)=B2(I,JJ)
606   CONTINUE
600   CONTINUE
      DO 61 I=1,N-NG
      DO 61 JJ=NG,J
61    BB(I,JJ)=B3(I,JJ-NG+1)
      DO 62 I=N-NG+1,KKK
      DO 62 JJ=1,NG-1
62    BB(I,JJ)=B4(I-N+NG,JJ)
      DO 63 I=N-NG+1,KKK
      DO 63 JJ=NG,J
63    BB(I,JJ)=B5(I-N+NG,JJ-NG+1)
      IF(M1.EQ.0) GOTO 64
      DO 65 I=KKK,K
      DO 65 JJ=1,NG-1
65    BB(I,JJ)=B6(I-K+1,JJ)
      DO 66 I=KKK,K
```

```
      DO 82 LJ=N-NG+1,KKK
      IG=I-2*NG
      IJ1=LJ-N+NG
82    CC(I,LJ)=C9(IG,IJ1)
      IF(M1.EQ.0) GOTO 83
      DO 84 I=1,NG
      DO 84 LJ=KKK+1,K
84    CC(I,LJ)=C6(1,LJ-KKK)
      DO 85 I=NG+1,2*NG
      DO 85 LJ=KKK+1,K
85    CC(I,LJ)=C7(I-NG,LJ-KKK)
      DO 86 I=2*NG+1,MMM
      DO 86 LJ=KKK+1,K
86    CC(I,LJ)=0.0
C     ###############################################################
C     ####### FORMATION OF [D1],[D2],[D3],[D4],[D5],[D6] #########
C     ###############################################################
83    DO 87 I=1,NG
      DO 88 LJ=1,NG
      IF(LJ.EQ.I) GOTO 89
      D4(I,LJ)=0.0
      D2(I,LJ)=ABS(V(I)*Y1(I,LJ))*COS(TH(I,LJ)+DEL(I)-DEL(LJ))
      IF(LJ.EQ.1) GOTO 91
      OLF=TH(I,LJ)+DEL(I)-DEL(LJ)
      D1(I,LJ-1)=ABS(V(I)*V(LJ)*Y1(I,LJ))*SIN(OLF)
91    GOTO 88
89    D1(I,LJ-1)=0.0
      D2(I,LJ)=0.0
      D4(I,LJ)=1.0
      DO 90 LK=1,N
```

```
      GOTO 71
72    C7(KJ,NK)=1.0
      C6(KJ,NK)=0.0
      DO 73 LK=1,N
      IF(KJ.EQ.LK) GOTO 73
      TKLI = TH(KJ,LK)+DEL(KJ)-DEL(LK)
      C6(KJ,NK)=ABS(V(LK)*Y1(KJ,LK))*COS(TKLI)+C6(KJ,NK)
73    CONTINUE
      C6(KJ,NK)=2*ABS(V(KJ)*Y1(KJ,KJ))*COS(TH(KJ,KJ))+C6(KJ,NK)
71    CONTINUE
70    CONTINUE
C     ###################################################################
C     #####################      FORMATION OF [CC]      #############
C     ###################################################################
69    DO 77 I=1,NG
      DO 77 LJ=1,N-NG
77    CC(I,LJ)=C2(I,LJ)
      DO 78 I=1,NG
      DO 78 LJ=N-NG+1,KKK
78    CC(I,LJ)=C3(I,LJ-N+NG)
      DO 79 I=NG+1,2*NG
      DO 79 LJ=1,N-NG
79    CC(I,LJ)=0.0
      DO 80 I=NG+1,2*NG
      DO 80 LJ=N-NG+1,KKK
80    CC(I,LJ)=0.0
      DO 81 I=2*NG+1,MMM
      DO 81 LJ=1,N-NG
81    CC(I,LJ)=0.0
      DO 82 I=2*NG+1,MMM
```

```
      IF(I.EQ.LK) GOTO 90
      TH2 =TH(I,LK)+DEL(I)-DEL(LK)
      D2(I,LJ)=ABS(V(LK)*Y1(I,LK))*COS(TH2)+D2(I,LJ)
      IF(LJ.EQ.1) GOTO 90
      THUI= TH(I,LK)+DEL(I)-DEL(LK)
      D1(I,LJ-1)=-ABS(V(I)*V(LK)*Y1(I,LK))*SIN(THUI)+D1(I,LJ-1)
90    CONTINUE
      D2(I,LJ)=2*ABS(V(I)*Y1(I,I))*COS(TH(I,I))+D2(I,LJ)
88    CONTINUE
87    CONTINUE
      IF(M1.EQ.0) GOTO 92
      DO 93 I=1,M1
      NL=PP(I)
      DO 94 LK = 1,NG
      D2(LK,NL)=D2(LK,NL+1)
      D4(LK,NL)=D4(LK,NL+1)
94    CONTINUE
93    CONTINUE
C     ##############################################################
C     ################           FORMATION OF [DD]        ############
C     ##############################################################
92    DO 95 I=1,NG
      DO 95 LJ=1,NG-1
95    DD(I,LJ)=D1(I,LJ)
      DO 96 I=1,NG
      DO 96 LJ=NG,J
96    DD(1,LJ)=D2(I,LJ-NG+1)
      DO 97 I=NG+1,2*NG
      DO 97 LJ=1,NG-1
```

```
      DO 66 JJ=NG,J
66    BB(I,JJ)=B7(I-K+1,JJ-NG+1)
C     ##########################################################
C      FORMATION OF [C2],[C3],[C4],[C5],[C6],[C7],[C8],[C9],[C10]
C     ##########################################################
64    DO 67 I=1,NG
      DO 68 JJ=NG+1,N
      J1J=JJ-NG
      HIJL=TH(I,JJ)+DEL(I)-DEL(JJ)
      C2(I,J1J)=ABS(V(I)*V(JJ)*Y1(I,JJ))*SIN(HIJL)
      C3(1,J1J)=ABS(V(1)*Y1(I,JJ))*COS(HIJL)
68    CONTINUE
67    CONTINUE
      DO 74 I=1,N-NG
      DO 75 LJ=1,N-NG
      IF (I.EQ.LJ) GOTO 76
      C9(I,LJ)=0.0
      GOTO 75
  76  C9(I,LJ)=1.0
  75  CONTINUE
  74  CONTINUE
      IF(M1.EQ.0) GOTO 69
      DO 70 I=1,M1
      NK=PP(I)
      DO 71 KJ=1,NG
      IF(KJ.EQ.NK) GOTO 72
      TH1 =TH(KJ,NK)+DEL(K)-DEL(NK)
      C6(KJ,NK)=ABS(V(KJ)*Y1(KJ,NK))* COS(TH1)
      C7(KJ,NK)=0.0
```

```
 97    DD(I,LJ)=0.0
       DO 98 I=NG+1,2*NG
       DO 98 LJ=NG,J
 98    DD(I,LJ)=D4(I-NG,LJ-NG+1)
       DO 99 I=2*NG+1,MMM
       DO 99 LJ=1,J
 99    DD(I,LJ)=0.0
C      ##############     [AA] =  [AA]   INVERSE        ################
       CALL MINE(AA,AA,K)
C      ##############     [CC] = [CC]   X   [AA]        ################
       CALL MTML(CC,AA,CC,M,K,K)
C      ##############     [EE] = [CC]   X   [BB]        ################
       CALL MTML(CC,BB,EE,M,K,J)
C      ##############     [GG] =  [DD]   -   [EE]       ################
       CALL SUB(DD,EE,GG,M,J)
C      ##############     [DD] =  [GG]                  ################

       DO 100 I = 1,M
       DO 100 LJ = 1,J
 100   DD(I,LJ) = GG(I,LJ)
C      ##############     [BB]  =  [AA]   X   [BB]      #################
       CALL MTML(AA,BB,BB,K,K,J)
C      ################     [FF]  =  [GG]  TRANSPOSE    #################
       CALL TRANS(GG,FF,M,J)
C      ################     [HH]  =  [FF]   X   [W]     #################
       CALL MTML(FF,W,HH,J,M,M)
C      ##############     [ALIT]  =   [HH]  X  [GG]  #################
       CALL MTML(HH,GG,ALIT,J,M,J)
C      ##############     UPPER   PART [GG]  CONTIANS  [ALIT] #######
       DO 101 I = 1,J
```

```
      DO 101 LJ = 1,J
101   GG(I,LJ) = ALIT(I,LJ)
C     ###############   [ALIT]  =  [ALIT]  INVERSE ##################
      CALL MINE(ALIT,ALIT,J)
      KQ=1
C     ###############   [TT]  =  [S]  -  [A]        ##################
      CALL SUB(S,A,TT,K,KQ)
C     ###############   [TT]  =  [AA]  X  [TT]      ##################
      CALL MTML(AA,TT,TT,K,K,KQ)
C     ###############   [TZ]  =  [T]  -  [B]        ##################
      CALL SUB(T,B,TZ,M,KQ)
C     ###############   [TQ]  =  [CC]  X  [TT]      ##################
      CALL MTML(CC,TT,TQ,M,K,KQ)
C     ###############   [TZ]  =  [TZ]  -  [TQ]      ##################
      CALL SUB(TZ,TQ,TZ,M,KQ)
C     ###############   [TYY]  =  [HH]  X  [TYY]    ##################
      CALL MTML(HH,TZ,TYY,J,M,KQ)
C     ###############   [DEY]  =  [ALIT]  X  [TYY] ##################
      CALL MTML(ALIT,TYY,DEY,J,J,KQ)
C     ###############   [ALY]  =  [BB]  X  [DEY]    ##################
      CALL MTML(BB,DEY,ALY,K,J,KQ)
C     ###############   [ALLT]  =  [AA]  X  [ALY]   ##################
      CALL MTML(AA,ALY,ALLT,K,K,KQ)
C     ###############   [DEX]  =  [TT]  -  [ALLT]  #################
      CALL SUB(TT,ALLT,DEX,K,KQ)
      DO 102 I=1,K+J
      IF(I.EQ.K) GOTO 113
      DEXY(I,1)=DEX(I,1)
      GOTO 103
```

```
 113    DEXY(I,1)=DEY(I-K,1)

 103    IF(DEXY(I,1).GT.TOL) GOTO 105

 102    CONTINUE

        GOTO 106

 105    DO 107 I=1,K

        X(I)=X(I)+DEX(I,1)

        IF(I.GT.J)GOTO 107

        Y(I)=Y(I)+DEY(I,1)

 107    CONTINUE

        DO 108 I=1,M1

        NJ=PP(I)

        Y(NG-1+NJ)=X(KKK+I)

 108    CONTINUE

2000    CONTINUE

 106    DO 109 I=1,NG

        PPP(I)=B(I,1)

        QP(I)=Q(I)

109     CONTINUE

        DO 110 I=1,KKK

        IF (I.GE.N-NG+1) GOTO 111

        PPP(NG+I)=A(I,1)

        GOTO 110

 111    QP(I-1)=A(I,1)

 110    CONTINUE

        WRITE(*,400)

 400    FORMAT(5X,'VOLTAGE',7X,'VOLTAGE ANGLE',7X,'      REAL',5X,
    1   '       REACTIVE')

        WRITE(*,401)

 401    FORMAT(22X,10HIN RADIANS,13X,5HPOWER,15X,5HPOWER)
```

```
      DO 112 I=1,N
      WRITE(*,206) V(I),DEL(I),PPP(I),QP(I)
112   CONTINUE
206    FORMAT(3X,F8.5,10X,F8.5,14X,F10.5,13X,F8.5)
199   FORMAT(I2,I2,F10.6,I2)
200   FORMAT(F8.5)
201   FORMAT(2F6.2)
202   FORMAT(5F8.5)
203   FORMAT(7F10.4)
204   FORMAT(10F7.3)
205   FORMAT(5F7.3)
500   FORMAT(/2X,8F10.5/)
      STOP
      END

C     ############# FORMING THE YBUS ########################
      SUBROUTINE YBUS(CZ,CH,Y1,TH,N)
      COMPLEX CH,CZ,CY,CA
      DIMENSION CZ(10,10), CA(10,10), CH(10,10), CY(10,10),Y1(10,10)
      DIMENSION TH(10,10)
      DO 7 I=1,N
      DO 7 LJ=1,N
      IF (CZ(I,LJ).EQ.0.0) GO TO 7
      CA(I,LJ)=1.0/CZ(I,LJ)
7     CONTINUE
      DO 25 I=1,N
      CY(I,I)=0.0
      DO 35 LJ=1,N
      IF (I.EQ.LJ) GOTO 35
```

```
      LR = I
      LC = J
120   CONTINUE
130   CONTINUE
      IF ( AL . EQ . 0.0 )  GO TO 270
      IF ( LR . EQ . IC . AND . LC . EQ .IC ) GOTO 160
      DO 140 I = 1,M
      D(I) = B(IC,I)
      B(IC,I) = B(LR,I)
      B(LR,I) = D(I)
140   CONTINUE
      DO 150 I = 1,N
      D(I) = B(I,IC)
      B(I,IC) = B(I,LC)
      B(I,LC) = D(I)
150   CONTINUE
      MC1(IC)= IC
      MC2(IC)= LC
160   IF( B(IC,IC) . EQ . 0.0 ) GO TO 270
      P = B(IC,IC)
      IF ( P . EQ . 1.0 ) GO TO 175
      DO 170 I = 1,M
      B(IC,I) = B(IC,I)/P
170   CONTINUE
175   DO 190 I = 1,N
      IF ( I . EQ . IC ) GO TO 190
      P1 = B(I,IC)
      DO 180 J = IC,M
      B(I,J) = B(I,J) - B(IC,J) * P1
```

```
      DO 5 I = 1,N
      MC1(I)= 0
      MC2(I)= 0
    5 CONTINUE
      M = N + N
      M1 = N + 1
      DO 70 I = 1,N
      DO 60 J = 1,N
      B(I,J) = A(I,J)
   60 CONTINUE
   70 CONTINUE
      DO 90 I = 1,N
      DO 80 J = M1,M
      B(I,J) = 0.0
   80 CONTINUE
   90 CONTINUE
      DO 100 I = 1,N
      M2 = I + N
      B(I,M2) = 1.0
  100 CONTINUE
      IC = 1
  110 AL = B(IC,IC)
      LR = IC
      LC = IC
      DO 130 I = IC,N
      DO 120 J = IC,N
      IF ( ABS (B(I,J) ) . LE . AL ) GO TO 120
      AL = B(I,J)
```

```
        CY(I,I)=CY(I,I)+CA(I,LJ)+CH(I,LJ)
 35     CONTINUE
 25     CONTINUE
        DO 40 I=1,N
        DO 50 LJ=1,N
        IF (I.EQ.LJ) GOTO 50
        CY(I,LJ)=-CA(I,LJ)
 50     CONTINUE
 40     CONTINUE
        DO 60 I=1,N
        DO 70 LJ=1,N
        Y1(I,LJ)=CABS(CY(I,LJ))
        OO=AIMAG(CY(I,LJ))
        ZZ=REAL(CY(I,LJ))
        IF(OO.EQ.0.0.AND.ZZ.EQ.0.0) GOTO 70
        IF(ZZ.EQ.0.0.AND.OO.NE.0.0) GOTO 4000
        RR=OO/ZZ
        TH(I,LJ)=ATAN(RR)
        GOTO 70
 4000   TH(I,LJ) =1.57
 70     CONTINUE
 60     CONTINUE
        RETURN
        END
C       ############### MATRIX INVERSE ##########################
        SUBROUTINE MINE(A,C,N)
        DIMENSION A(8,8) , C (8,8) ,          16)
        DIMENSION MC1 (8) , MC2 (8) ,       (16)
```

```
180     CONTINUE
190     CONTINUE
        IF ( IC . EQ . N ) GO TO 200
        IC = IC + 1
        GO TO 110
200     DO 220 I = 1,N
        DO 210 J = 1,N
        J1 = J + N
        C(I,J) = B(I,J1)
210     CONTINUE
220     CONTINUE
        DO 240 IQ = N,1,-1
        K1=MC1(IQ)
        K2=MC2(IQ)
        IF ( K1 . EQ . 0 ) GO TO 240
        DO 230 J = 1,N
        D(J) = C(K1,J)
        C(K1,J) = C(K2,J)
        C(K2,J) = D(J)
230     CONTINUE
240     CONTINUE
        GO TO 290
270     WRITE (*,280)
280     FORMAT ( 5X,'INVERSE IS NOT EXISTING')
290     CONTINUE
        RETURN
        END

C       ############# MATRIX MULTIPLICATION ########################
```

INPUT DATAS

```
0502000.00100005
00.00000
000.50001.00
01.0600001.0470001.0181001.0225001.0240
00.00000-0.04897-0.10740-0.09300-0.08720
01000.000001000.000001000.000001000.000001000.000001000.000001000.0000
-1000.0000-1000.0000-1000.0000-1000.0000-1000.0000-1000.0000-1000.0000
   .000   .000   .020   .000   .000   .000   .000   .000   .080   .240
   .020   .060   .000   .000   .040   .120   .060   .180   .060   .180
   .000   .000   .040   .120   .000   .000   .080   .240   .000   .000
   .000   .000   .060   .180   .080   .240   .000   .000   .010   .030
   .080   .240   .060   .180   .000   .000   .010   .030   .000   .000
   .000   .000   .000   .030   .000   .000   .000   .000   .010   .025
   .000   .030   .000   .000   .000   .015   .000   .020   .000   .020
   .000   .000   .000   .015   .000   .000   .000   .025   .000   .000
   .000   .000   .000   .020   .000   .025   .000   .000   .000   .010
   .000   .025   .000   .020   .000   .000   .000   .010   .000   .000
001.000001.000001.000001.000001.000
```

```
      DO 1 I = 1,L
      DO 1 K = 1,N
      B(K,I)=A(I,K)
1     CONTINUE
      RETURN
      END
C     ############### MATRIX ADDITION ########################
      SUBROUTINE ADD(A,B,C,L,N)
      DIMENSION A(8,8),B(8,8),C(8,8)
      DO 1 I = 1,L
      DO 1 K = 1,N
      C(I,K) =  A(I,K) + B(I,K)
1     CONTINUE
      RETURN
      END
C     ############### MATRIX SUBTRACTION #######################
      SUBROUTINE SUB (A,B,C,L,N)
      DIMENSION A(8,8),B(8,8),C(8,8)
      DO 1 I = 1,L
      DO 1 K = 1,N
      C(I,K) = A(I,K) - B(I,K)
1     CONTINUE
      RETURN
      END
C     ################# SUBROUTINES OVER   ############################
```

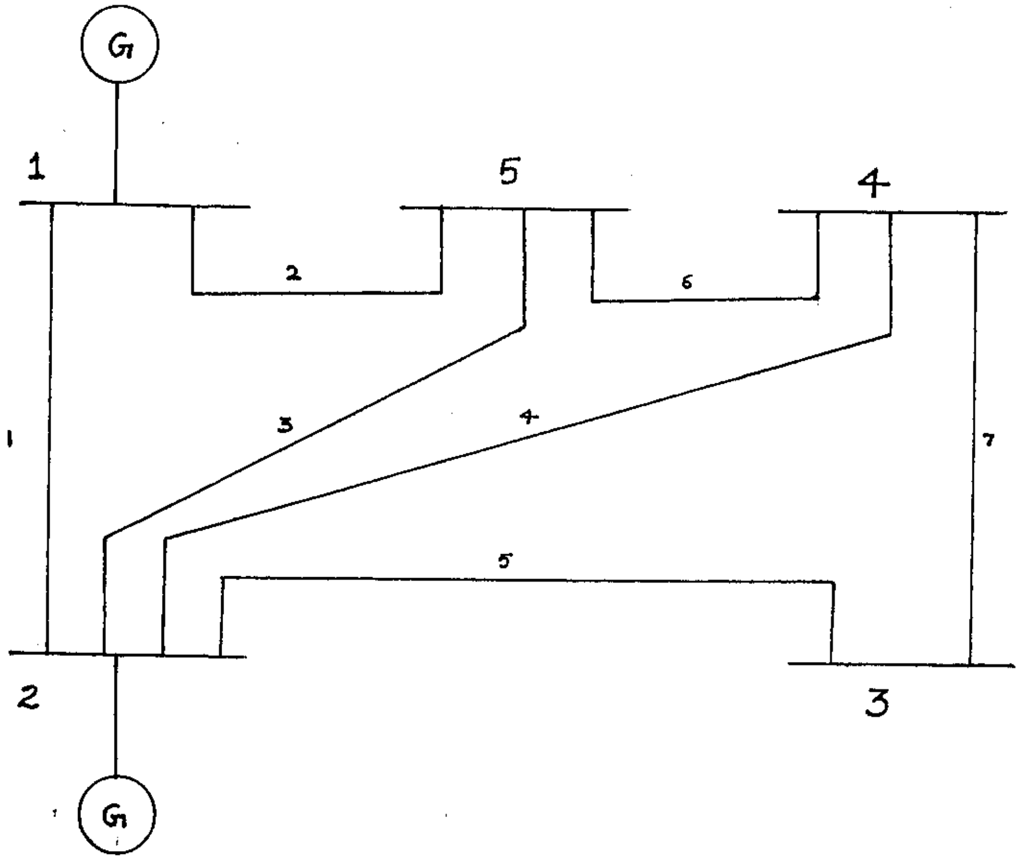
```
      SUBROUTINE MTML(A,B,C,L,M,N)
      DIMENSION A(8,8) , B(8,8) , C(8,8)
      DO 1 I=1,L
      DO 2 J=1,N
      C(I,J)=0.0
      DO 3 K=1,M
      C(I,J)=C(I,J)+A(I,K)*B(K,J)
3     CONTINUE
2     CONTINUE
1     CONTINUE
      RETURN
      END
C     ############# CALCULATING  P  AND  Q  ######################
      SUBROUTINE NR(V,DEL,Y1,TH,N,P,Q)
      DIMENSION P(5),Q(5),Y1(10,10),TH(10,10),DEL(5),V(5)
      DO 1 I = 1,N
      P(I)=0.0
      Q(I)=0.0
      DO 2 K = 1,N
      P(I)=ABS(V(I)*V(K)*Y1(I,K))*COS(TH(I,K)+DEL(I)-DEL(K))+P(I)
      Q(I)=-ABS(V(I)*V(K)*Y1(I,K))*SIN(TH(I,K)+DEL(I)-DEL(K))+Q(I)
2     CONTINUE
1     CONTINUE
      RETURN
      END
C     ################### FINDING TRANSPOSE ########################
      SUBROUTINE TRANS(A,B,L,N)
      DIMENSION A(8,8),B(8,8)
```

RESULTS

```
                                                      REAL              REACTIVE
VOLTAGE       VOLTAGE ANGLE                              POWER              POWER
                   IN RADIANS                       -1.02600              .67194
1.04600              .00000                          -.00009              .22250
1.04320            -.04590                            .45000             -.46000
1.01800            -.10793                            .33950             -.29130
1.02530            -.09300                            .30490             -.39153
1.02140            -.08610
```

formula used

FIVE BUS TEST SYSTEM



BUS DATA

Bus Number	Bus Code	Initial Bus voltage		Generation		Load	
		Mag P.U	Angle P.U	MW	MUAR	MW	MUAR
1*	1	1.060	0	0	0	0	0
2	2	1.047	0	40	30	20	10
3	3	1.000	0	0	0	60	10
4	4	1.000	0	0	0	40	5
5	5	1.000	0	0	0	45	15

conclusion

CONCLUSION

A new concept involving an optimal approach to the use of load flow as a planning tool is presented in this work. Normally at the planning stage, the system planner has to carry out a number of successive load flow studies, each time varying the specification, before an acceptable operating condition is attained. The success of this procedure depends almost entirely on the skill and experience of the planner and his knowledge of the system. The new approach suggested here eliminates this drawback and uses an optimisation technique to arrive at the best solution in a single run of the program.

The method was implemented using a Fortan program on a sample 5 bus (thee IEEE standard) system. Convergence was obtained in 3 iteration to a tolerance of 0.001 per unit. The results were very encouraging and the authors feel reasonably certain that the method can be applied with success to large systems also.

The main aim of the present work was to illustrate the validity of the new approach. The computations effort required is more than for a conventional load flow and further work needs to be carried out to improve the efficiency and speed of the program.

bibliography

BIBLIOGRAPHY

1. **"COMPUTER METHODS IN POWER SYSTEM ANALYSIS"**

   BY STAGG AND EI _ ABIAD.

2. **"ELECTRICAL POWER SYSTEMS**

   BY C.L. WADHWA

3. **"JOURNAL OF THE INTSITUTION OF ENGINEERS (India)"**

   VOLUME 69, AUGUST 1 988