A LOAD FLOW AND ECONOMIC DISPATCH COMPUTER CODE FOR THE TUCSON GAS AND ELECTRIC POWER SYSTEM

by

Richard Francis Baldwin

A Thesis Submitted to the Faculty of the

DEPARTMENT OF ELECTRICAL ENGINEERING

In Partial Fulfillment of the Requirements
For the Degree of

MASTER OF SCIENCE

In the Graduate College

THE UNIVERSITY OF ARIZONA

1979

STATEMENT BY AUTHOR

This thesis has been submitted in partial fulfillment of requirements for an advanced degree at The University of Arizona and is deposited in the University Library to be made available to borrowers under rules of the Library.

Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgment of source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by the head of the major department or the Dean of the Graduate College when in his judgment the proposed use of the material is in the interests of scholarship. In all other instances, however, permission must be obtained from the author.

SIGNED: Richard P Baldwin

APPROVAL BY THESIS DIRECTOR

This thesis has been approved on the date shown below:

Robert N. Carlile
ROBERT N. CARLILE
Professor of Electrical Engineering

Jan 17, 1978
Date

TABLE OF CONTENTS

Page

LIST OF ILLUSTRATIONS v

LIST OF TABLES vi

ABSTRACT vii

CHAPTER

1. INTRODUCTION 1

Background 2

2. COMPUTER CONSIDERATIONS 7

Interactive Capabilities 8
Temporary Files 9
User Limitations 10

3. POWER SYSTEM MODEL 12

Transmission Line Characteristics 13
The Load Flow Problem 19
The Economic Dispatch Problem 27

4. LOAD FLOW ANALYSIS 38

Solution Method and Flow Chart 49

5. ECONOMIC DISPATCH ANALYSIS 53

Solution Method and Flow Chart 54

6. TGE POWER SYSTEM 58

TGE Line Data 61
TGE Bus Data 62
Generator Data 62

7. COMPUTER CODE DESCRIPTION AND OPERATION . . . 66

Diagnostics 72

TABLE OF CONTENTS--*Continued*

Page

8. RESULTS AND CONCLUSIONS 75

Results 75
Conclusions 78
Recommendations 79

APPENDIX A. SAMPLE PROBLEMS 81

APPENDIX B. COMPUTER CODE LISTINGS 118

REFERENCES 184

LIST OF ILLUSTRATIONS

Figure Page

2.1 Computer Code Memory Requirements 11

3.1 Transmission Line Equivalent Circuit 14

3.2 Typical 4-Bus Power System 17

3.3 Typical Generator Cost Curve 29

3.4 Cost and Incremental Cost Curves for Typical 4-Bus System 31

3.5 Graphical Economic Dispatch Solution 37

4.1 Load Flow Solution Flow Chart 52

5.1 Economic Dispatch Solution Flow Chart 57

6.1 TGE Transmission System 59

LIST OF TABLES

Table | Page

2.1 Temporary Computer Files 9

3.1 Typical Transmission Line Parameters 15

3.2 Line Data 18

3.3 Bus Data 26

3.4 Generator Data and Constraints 32

4.1 Bus Characteristics 43

6.1 Typical Generator Incremental Heat Rates . . . 63

ABSTRACT

A computer model of the Tucson Gas and Electric (TGE) power system and a computer code for the analysis of electrical power load flow and economic dispatch through the system was developed for use with the Digital Equipment Corporation System 10 (DEC-10) computer. The DEC-10 provides fast direct on-line response from remote terminals and user timesharing of its large memory storage capacity. The computer code was developed to demonstrate the cost advantages available through computerized control of electrical power distribution in large complex systems such as that owned by the TGE. Although preliminary data of distribution system parameters was provided by TGE, actual system cost and operating data were not available. System transmission lines, power source, bus, and load parameters can be varied to permit analysis of their effect on power flow and system operating costs. System size can be varied from a simple two or three bus system for instructional use to a large complex system such as that serving the Tucson metropolitan area with approximately 50 buses and 100 transmission lines. A unique Newton-Raphson computer code was developed for the complex system load flow calculations to obtain solutions with minimum computational iterations.

CHAPTER 1

INTRODUCTION

The use of digital computers in the analysis of electrical power system performance is well established (see, e.g., Happ, 1977; Sasson et al., 1971; Stott, 1974; Subramanian and Ramachandra Rao, 1969). However, little effort has been made to develop an interactive code to take advantage of current high speed, timesharing computers with large memory storage capabilities. This thesis presents such an interactive code for use with the Digital Equipment Corporation's System 10 (DEC-10) computer installed at The University of Arizona. The code was designed for the analysis of electrical power system performance through economic dispatch and load flow studies. Further, it was designed to give the power systems engineer maximum flexibility and control of computerized solution routines with minimum preparation and response times. In addition, through use of the DEC-10's remote station and timesharing capabilities the code can be used for instructional and demonstration purposes.

The computer code incorporates a unique algorithm for solution of the set of nonlinear load flow equations and uses an iterative Newton-Raphson solution method. Two codes were developed. EDSP15 was designed for instructional and demonstration use on small systems with no more than 15 buses and 30 transmission lines. This code also includes a successive displacement or Gauss-Seidel iterative solution method for comparison with the more rapid, reliable quadratic convergence rate of the Newton-Raphson method. The other code, EDSP50, is for use with systems serving large metropolitan areas such as Tucson, Arizona and can accept up to 50 buses and 100 transmission lines.

To demonstrate use of the computer code, a model of the Tucson Gas and Electric Company (TGE) power distribution system was developed. Input data used by the code to describe the system physical characteristics were derived and typical operating conditions were analyzed. Unfortunately, cost comparisons with actual operating condition at TGE could not be accomplished since their actual operating and cost data were not available.

The characteristics, capabilities, and limitations of the DEC-10 computer necessary for use of the code are included. Sample problems are also included to familiarize the user with the DEC-10 and the code operation.

Background

The primary function of any electrical power system is to provide the power required by the various loads connected to the system. Simultaneously, the generated real and reactive power, the bus voltages, and the voltage phase angles throughout the system must be kept within constraints imposed by equipment and system operating limits, while the load demands undergo large, and sometimes unpredictable, changes. Therefore, the most important or certainly the most frequently performed power system study is "load flow" where bus voltages and phase angles are calculated for various transmission network and load configurations.

Prior to the advent of large capacity digital computers, system load flow studies were performed by simulation techniques employing an analog A/C network analyzer. Ward and Hale (1956) provided one of the first practical automatic solution methods for use with the digital computer employing nodal admittance analysis and a Gauss-Seidel iterative process to solve the set of nonlinear equations involved. Improvements in solution techniques developed rapidly during the late '50s and early '60s. Among the more significant developments were: Van Ness (1959) proposed the use of Newton's methods for a more reliable and faster solution convergence; Tinney and Hart (1967) and Tinney and Walker (1967) developed techniques to drastically reduce computation time and memory storage requirements in solutions of very large systems, i.e., in excess of 1000 buses. Other efforts have been directed toward further improvements in speed and reliability of convergence, solution accuracy, reduction of memory storage requirements, and expansion of problem size through algorithms using diakoptics, tearing, and block-iteration techniques (see, e.g., Stott, 1974; Subramanian and Ramachandra Rao, 1969).

Solution of the load flow problem determines the real and reactive power flow, the bus voltages, and the voltage phase angles required to meet each bus or terminal condition, while economic dispatch is concerned with development of an optimum generation strategy to meet the load demands at a minimum or optimum system operating cost. Since there is an infinite number of generation profiles that could meet a specified load requirement, a method must be devised to find the least cost schedule while observing all the equipment and system constraints such as maximum generator or transformer ratings, bus voltage and system reactive power limits.

Prior to the 1930s, various methods were employed to divide the load among the available generation units for the minimum system operating cost. For example: the "Base Load" method loaded the most efficient unit to its maximum capacity first, then loaded the second most efficient unit to its capacity, etc., or the "Best Point Loading" method where units are successively loaded to their lowest heat rate point beginning with the most efficient unit and working down to the least efficient, etc. During the early

1930s, it was recognized that for the most economical operation, the incremental cost of all machines should be equal. In the early 40s, Steinberg and Smith (1943) published their classic book on economic loading of power plants that added understanding and impetus to the equal incremental method. In their approach, the generating unit's incremental cost characteristics were represented by straight line segments for easy applications. Also, George (1943) provided a breakthrough in system transmission loss calculations through the development of his loss formulas. By 1954, a transmission loss penalty computer was developed and used in conjunction with an incremental loading slide rule by American Electrical Power (AEP) to produce daily generation schedules. In 1955, an electronic differential analyzer was developed by Morrill and Blake (1955) for production of daily generation schedules. Early in the '60s, Carpentier (1962a, 1962b) provided a breakthrough in mathematical formulation of the optimum power flow problem using the Kuhn and Tucker (1951) theorems for nonlinear programming and a Gauss-Seidel solution procedure. Later, Dommel and Tinney (1968) supplied a significant advance in digital computer solution techniques by using Newton's methods for the load flow solutions and incorporating a steepest descent or gradient method for finding the optimum or most economical generation profile. They also employed the more lenient "penalty factors" instead of hard limits for equipment and system constraints and thereby enhanced convergence probability. Current efforts are being extended to system security analysis, reactive power scheduling and allocation, pollution dispatching, maximum interchange, expansion planning, maintenance scheduling, substation switching, and on-line automatic dispatching and control (see, e.g., Sasson and Merrill, 1974).

The initial effort at The University of Arizona to develop an interactive computer code for the solution of economic dispatch problems was developed for use with the PDP-9 mini-computer by Martinez (1973). This code would accommodate a 15 bus power system and used the Gauss-Seidel iterative solution method. Later Stanley Beaver adapted it for use on the DEC-10. This thesis develops a new computer code to take advantage of the large core memory and disc storage capability of the DEC-10 and will accommodate power systems as large as 50 buses and 100 transmission lines. Further, the new code incorporates an iterative Newton-Raphson solution method to provide more rapid and reliable convergence of the solution.

CHAPTER 2

COMPUTER CONSIDERATIONS

The Digital Equipment Corporation System 10 (DEC-10) computer used by the EDSP codes is a third generation, high speed computer with a large capacity central (core) memory of 524K words. It has a model KL10 central processor (CPU) controlled by the timesharing, TOPS-10 operating system. The CPU uses a 36 bit word size and includes double precision arithmetic instructions. Under the TOPS-10 monitor, interactive service for up to 90 customers is provided simultaneously. Fast access disk storage of over one billion characters is available for temporary storage of user files. In addition, three types of magnetic tape can be used with the DEC-10; seven-track, nine-track, or DECtape. DECtape is ideally suited for permanent storage of source codes or low volume data files. Each DECtape reel will store 22 files or approximately 353K characters whichever occurs first. DEC Model VT05, Teleray, or Infoton terminals are available for video input/output display or DECwriter terminals are available for printed (hard copy) input/output products. Remote station operation is available at numerous locations throughout the campus or operation can be set up at any location off campus that has normal telephone service. An acoustical coupler would be required to connect the terminal to any normal telephone instrument. Several telephone numbers are available for computer service on a 24 hour per day basis. An inter-computer link permits routing of output products to the high speed line printers, card or paper punch machines, or the Calcomp 12 inch plotter. Batch processing is also available for off-line execution of very large programs. Several text editing routines are available for direct input or modification of user files. The most popular and versatile is the SOS (Son of Stopgap) text editor.

Interactive Capabilities

The interactive capability of the code in conjunction with the DEC-10 offers a distinct advantage in reduced problem setup time, and while it is not necessary to maintain a constant dialogue with the computer, the system engineer is free to make decisions during the computational process and implement changes to optimize results without fear of computer hang-ups. Further, data and solution printouts can be controlled to meet specific needs. Data input and modification is straightforward and requires no card punching or formating. During each run the EDSP code will prompt the user when action or data input is required. The specific prompting messages and the response required is outlined in Chapter 7.

Temporary Files

As stated above, disk storage is available for temporary data files so that subsequent runs of the EDSP code can be made without re-entering all input data. Although the temporary files can be created separately through use of the DEC-10 SOS text editor, the code automatically creates the temporary files as the data are being entered during program execution. Since disk storage is for common use by all users, large data files such as the Y and Z bus matrix, discussed in Chapter 3, should be deleted prior to "logging off" the terminal or they may be transferred to DECtape for permanent storage and recalled for use at a later date. The temporary files created during program execution are shown in Table 2.1.

Table 2.1 Temporary Computer Files

Type Data	File Name
Line Data	FOR20.DAT
Bus Data	FOR21.DAT
Generator Data	FOR22.DAT
Generator Constraints	FOR23.DAT
Y/Z Bus Matrix	FOR24.DAT
Problem Solutions[a]	SOL.DAT

[a] Temporary file of the problem solutions is optional and must be requested during program execution.

User Limitations

Users are normally limited to 60 pages (30K words) of central (core) memory use and 500 blocks (61K words) of disk storage while "logged in" and 300 blocks (36.6K words) with "logging off." The difference between "logged on" and "logging off" limits is to permit expanded use of disk storage while programs are being run or edited. Increased core and disk limits are available for large problems such as the TGE system but operation may be restricted to other than peak student use periods. Memory storage requirements to run the computer code EDSP50 are 188 pages of central (core) memory and 540 blocks of disk memory. The core memory requirements vary with the number of buses in the system as shown in Figure 2.1.

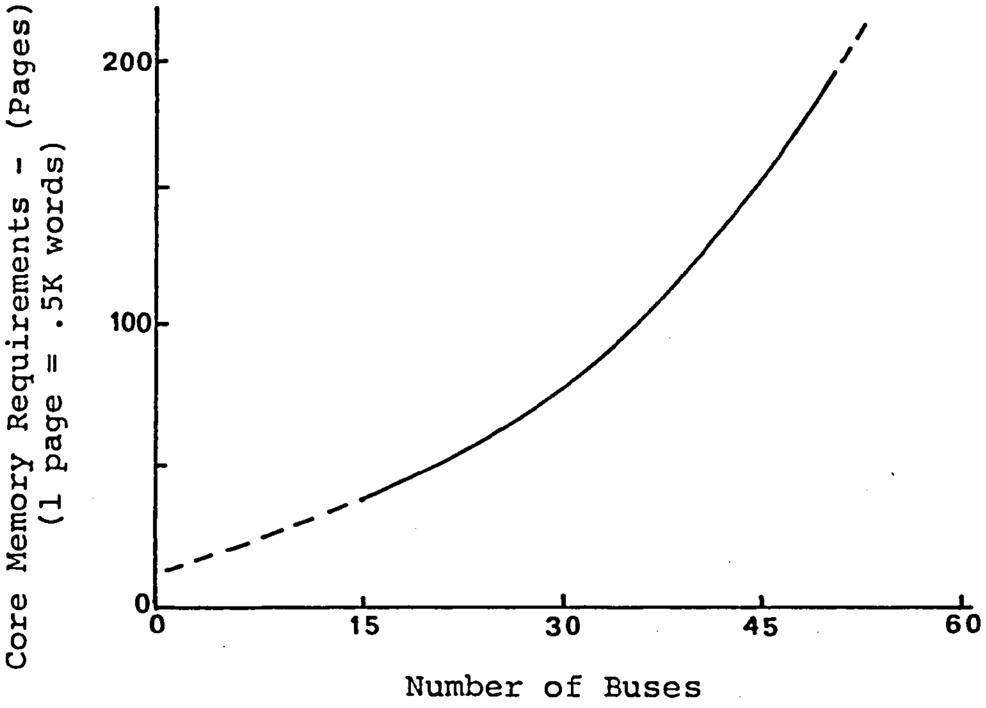


Figure 2.1 Computer Code Memory Requirements

CHAPTER 3

POWER SYSTEM MODEL

Most power systems consist of a generation capability, a distribution network which transports electrical power to the customer loads, and a monitoring and control system to operate the system under normal or emergency conditions. The objective of this chapter is to obtain a mathematical model of a power system for study of electrical power distribution and generation under normal operating conditions. Mathematical expressions are derived to describe the physical characteristics of the transmission line network, the bus or power load centers, the generator operating costs, and the system and equipment operating limits.

In developing the mathematical model, a nodal admittance form was selected since this form is more economical in the use of computer time and it requires less memory storage (see, e.g., Stagg and El-Abiad, 1968:258). In addition, the mathematical notation closely follows the transmission line network topography, making it easier to apply to computer programming techniques. Further, since only normal operating conditions are considered, a balanced three phase power system with a constant 60Hz frequency is assumed.

Transmission Line Characteristics

In a normal balanced three-phase system, the transmission line characteristics of each phase are considered identical. Thus, the computer model represents a single phase of the three phase system, i.e., the positive sequence network. Each transmission line used in the system can be represented by the following four electrical parameters per unit length of line: series resistance (R_i), series reactance (X_i), shunt conductance (G_i), and shunt susceptance (B_i). In the metropolitan areas served, the transmission lines are usually less than 100 miles long and consequently are considered "electrically short." Thus, the distributed electrical parameters can be "lumped" into an equivalent π circuit as shown in Figure 3.1 (see, e.g., Elgerd, 1971:193). In this equivalent circuit, the series impedance (z_{ik}) is complex,

$$z_{ik}\text{ser} = z_{ki}\text{ser} = (R + jX)\ell \text{ ohms} \tag{3.1}$$

R = line resistance per unit length in ohms

X = line inductive reactance per unit length in ohms

ℓ = line length in feet, meters, or miles

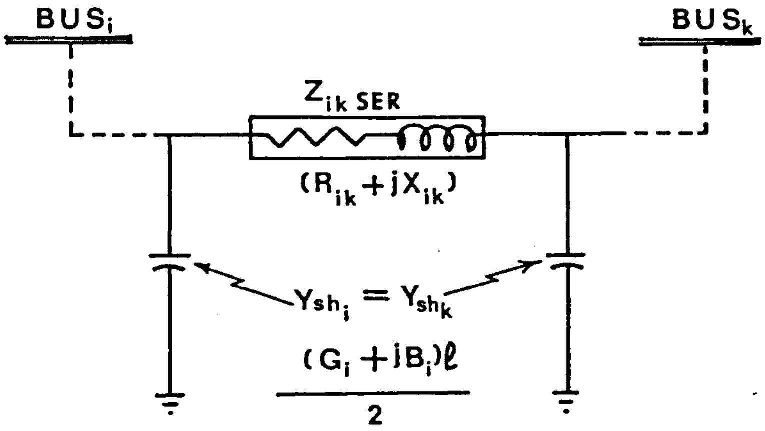


Figure 3.1 Transmission Line Equivalent Circuit

The equivalent shunt admittance is also complex and one-half of the total distributed admittance is applied to each leg of the equivalent π network.

$$Y_{shi} = Y_{shk} = \frac{(G + jB)}{2}\ell \text{ mhos} \tag{3.2}$$

G = line conductance per unit length in mhos

B = line susceptance per unit length in mhos

ℓ = line length in feet, meters, or miles

As shown by Elgerd (1971:170), the series elements, of which the inductive reactance is often dominant, set the limit of the current flowing through the line and thereby determine the power transmittability or capacity of the transmission line. The series resistance is usually much smaller than the series reactance (see, e.g., Table 3.1) but

Table 3.1 Typical Transmission Line Parameters

Type Conductors	Shunt admittance (B+jG) mhos/1000'	Series Impedance (R+jX) ohms/1000'
Copper #4/0	0.0 + j0.0	0.0574 + j0.1282
Aluminum AA #477MCM	0.0 + j0.0	0.0409 + j0.1195
Aluminum ACSR #366MCM	0.0 + j0.0	0.0580 + j0.1196
Aluminum AA #750 UG	$0.0 + j4.3731\text{x}10^{-5}$	0.0320 + j0.0380

is important in that it determines the system real losses. The shunt elements represent a "leakage" path for the line current between the phases and ground. These leakage currents are influenced by the geometry of the transmission line, the physical characteristics of the transmission cable, and are proportional to the line voltage. The shunt conductance G_i is extremely small and, if leakage occurs at all, it usually takes place along the insolator strings. It is strongly affected by the weather, atmospheric conditions, and the salt content in the air. It is customary to neglect this parameter under normal operating conditions. The shunt susceptance B_i is also negligible except for underground cables and overhead transmission lines where the line voltage exceeds 300KV and/or the line length exceeds 200 miles. Typical values for these parameters are shown in Table 3.1 for the type lines indicated. The EDSP computer code requires these line parameters to be entered as "Line Data" in per unit values per unit length of transmission line. For example, consider the 4-bus power system shown in Figure 3.2. The typical line parameters were used as indicated and are shown in Table 3.2 in per unit values required by the code. The impedance base Z_b used for the system was 21.16 ohms, i.e., $Z_b = V_b^2/S_b$. It should be noted that a "starting bus" or "ending bus" shown in Table 3.2 does not exist in an actual system. These terms are used for the expediency of describing components of current and power in particular transmission lines and should not be interpreted as limiting the actual flow of current or power to a particular direction. Also, the numbering of buses and transmission lines is completely arbitrary and used to facilitate network description only.

Transformers used to step-up or step-down the level of voltage between buses are included by the computer code as transmission lines. Since the transformer characteristics can be represented by an equivalent π network like the one shown in Figure 3.1 for transmission lines (Elgerd, 1971:126), the appropriate equivalent impedance and admittance values are included by the code as "Line Data" to represent these transformer parameters. Problem #2 in Appendix A includes an example of how transformers are represented in the system.

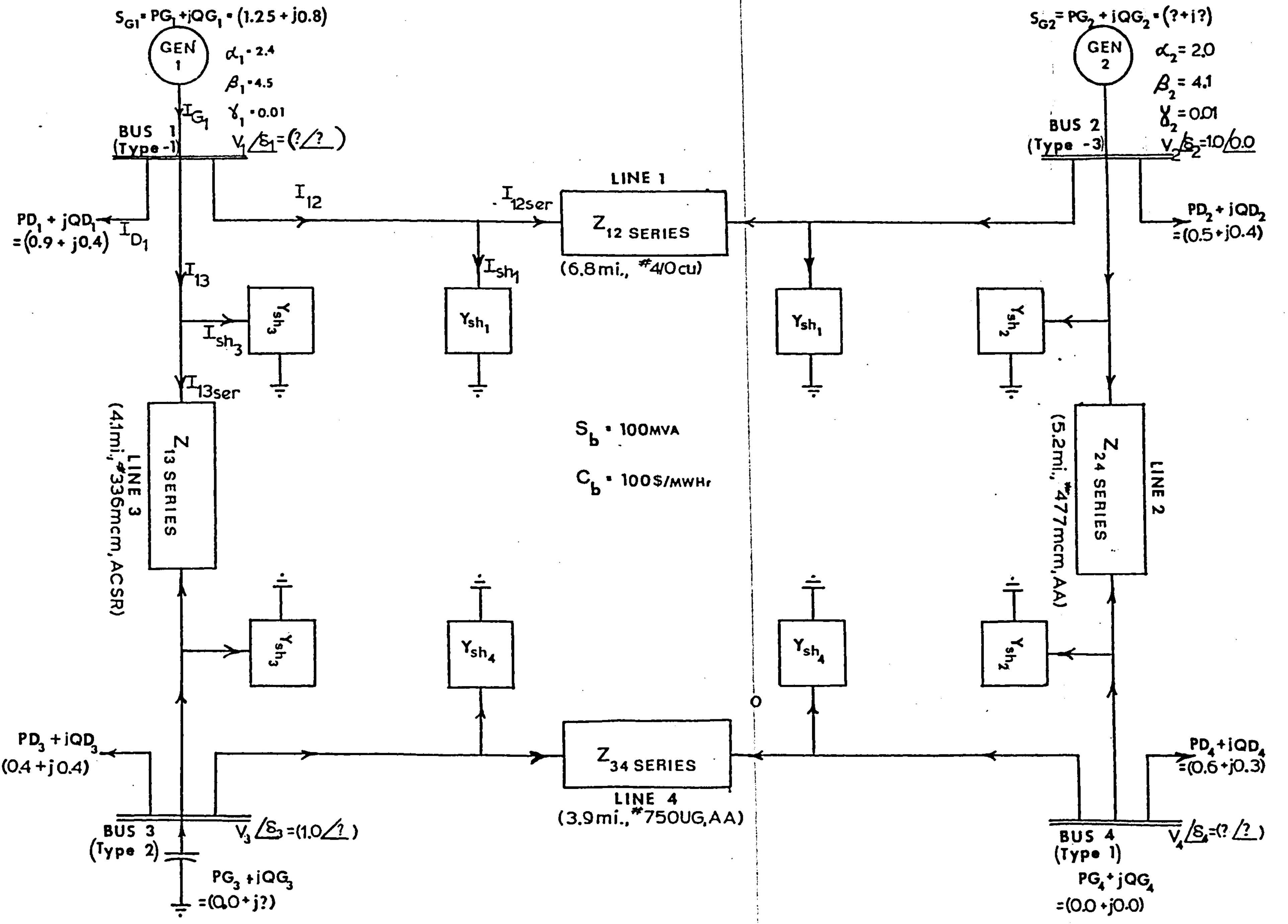


Figure 3.2 Typical 4-Bus Power System

Table 3.2 Line Data

Line No.	Starting Bus	Ending Bus	Length (1000')	Shunt Admittance (G+jB) per unit	Series Impedance (R+jX) per unit
1	1	2	35.904	0.0 + j0.0	$0.27127\text{x}10^{-2} + j0.6059\text{x}10^{-2}$
2	2	4	27.456	0.0 + j0.0	$0.19329\text{x}10^{-2} + j0.5648\text{x}10^{-2}$
3	1	3	21.648	0.0 + j0.0	$0.2741\text{x}10^{-2} + j0.5652\text{x}10^{-2}$
4	3	4	20.592	$0.0 + j9.2531\text{x}10^{-4}$	$0.15123\text{x}10^{-2} + j0.17958\text{x}10^{-2}$

The Load Flow Problem

In the nodal admittance model used to represent the power system, nodes are buses or load centers to which the various loads and generators are connected. Transmission lines interconnect these nodes (buses) to transport power between buses as required. Power provided by the generators S_{Gi} and power consumed by the loads S_{Di} at bus i are complex. That is, they have the following real and reactive components: $S_{Gi} \equiv PG_i + jQG_i$ and $S_{Di} \equiv PD_i + jQD_i$. To establish a sign convention, power into a node was considered positive. Also, phasor quantities are represented by capital letters. Thus, the net injected complex power S_i available at bus i is defined as the difference between the generated power input and the power consumed by the loads.

$$S_i \equiv P_i + jQ_i \equiv (PG_i - PD_i) + j(QG_i - QD_i) \qquad (3.3)$$

Any power shortage or excess at each bus must be transported by the interconnecting transmission lines. These line powers can be expressed in terms of the bus voltages and currents as follows. Using phasor notation, let V_i or $|V_i|e^{j\delta_i}$ represent the ith bus phase to ground voltage with the magnitude $|V_i|$ obtained with respect to the reference or ground node and the phase angle δ_i representing the angle between the reference "slack" bus voltage and the ith bus voltage. Also, let I_i or $|I_i|e^{j\psi_i}$ represent the net injected current (current into a bus is considered positive) at bus i with ψ_i representing the angle between the reference "slack" bus voltage and the ith bus current. Then the net complex bus power S_i at bus i is defined as:

$$S_i \equiv P_i + jQ_i \equiv V_i I_i^* = |V_i| e^{j\delta_i} |I_i| e^{-j\psi_i}$$
$$= |V_i||I_i| e^{j\theta_i} \tag{3.4}$$

with the complex power angle $\theta_i = (\delta_i - \psi_i)$. Also, in a nodal analysis, the currents entering and leaving the node must obey Kirchhoff's current law, that is, their algebraic sum must equal zero. As an example, consider bus 1 in Figure 3.2. It can be seen that the generated current I_{Di} that is not consumed in the load I_{Di} must flow through the transmission lines 1 and 3, or as stated above $I_{Gi} - I_{Di} - I_{12} - I_{13} = 0$. Clearly, the net injected current at bus 1 is:

$$I_1 = I_{G1} - I_{D1} = I_{12} + I_{13} = V_1 y_{sh1} + \frac{V_1 - V_2}{z_{12ser}}$$
$$+ V_1 y_{sh3} + \frac{V_1 - V_3}{z_{13ser}} \tag{3.5}$$

This equation can be rewritten in terms of the bus powers and voltages using (3.4).

$$\frac{S_1^*}{V_1^*} = V_1(y_{sh1} + y_{sh3}) + (V_1 - V_2)y_{12} + (V_1 - V_3)y_{13} \tag{3.6}$$

where from Equation (3.1),

$$y_{ik} = \frac{1}{z_{ikser}} = \frac{1}{R + jX} \text{ mhos} \tag{3.7}$$

Thus, the power flow equations for the 4-bus system in Figure 3.2 are:

$$S_1{}^* = P_1 - Q_1 = V_1{}^*[(y_{sh1}+y_{sh3}+y_{12}+y_{13})V_1 - y_{12}V_2 - y_{13}V_3] \tag{3.8}$$

$$S_2{}^* = P_2 - Q_2 = V_2{}^*[(-y_{12}V_1+(y_{sh1}+y_{sh2}+y_{12}+y_{24})V_2 - y_{24}V_4] \tag{3.9}$$

$$S_3{}^* = P_3 - Q_3 = V_3{}^*[(-y_{13}V_1+(y_{sh1}+y_{sh4}+y_{13}+y_{34})V_3 - y_{34}V_4] \tag{3.10}$$

$$S_4{}^* = P_4 - Q_4 = V_4{}^*[(-y_{24}V_1 - y_{34}V_3+(y_{sh2}+y_{sh4}+y_{24}+y_{34})V_4] \tag{3.11}$$

Now the general form of the static load flow equation (SLFE) becomes apparent.

$$S_i{}^* = V_i{}^*I_i = V_i{}^* \sum_{k=1}^{n} V_k Y_{ik} \quad \begin{array}{l} i = 1,2,\ldots n \\ n = \text{number of buses} \end{array} \tag{3.12}$$

In this equation [Y] is defined as the complex admittance matrix representing the transmission line parameters in the distribution network. The diagonal elements $|Y_{ii}|e^{j\theta_{ii}}$ of this matrix are the sum of all shunt and series admittances terminating at bus i. The off-diagonal elements $|Y_{ik}|e^{y\theta_{ik}}$ are equal to the negative of the series admittance y_{ik} between buses i and k. The angle θ_{ik} is the admittance angle of the resulting elements. As an example, the admittance matrix for the 4-bus system in Figure 3.2 would be:

$$[Y] = \begin{array}{cccc} (y_{sh1}+y_{sh3} & -y_{12} & -y_{13} & 0 \\ +y_{12}+y_{13}) & & & \\ -y_{21} & +(y_{sh1}+y_{sh2} & -0 & -y_{24} \\ & +y_{21}+y_{24}) & & \\ -y_{31} & 0 & +(y_{sh1}+y_{sh3} & -y_{34} \\ & & +y_{31}+y_{34}) & \\ 0 & -y_{42} & -y_{43} & +(y_{sh2}+y_{sh4} \\ & & & +y_{42}+y_{43}) \end{array}$$

This is a symmetrical matrix of complex elements and in its general form is:

$$[Y] \equiv \begin{array}{ccc} Y_{ii} & \cdots & Y_{in} \\ \vdots & \ddots & \vdots \\ Y_{ni} & \cdots & Y_{nn} \end{array} \quad \begin{array}{l} i=1,2,\ldots n \\ k=1,2,\ldots n \\ n=\text{number of buses} \end{array} \qquad (3.13)$$

Equation (3.12) can be rewritten into two real equations as follows:

$$P_i = PG_i - PD_i = \sum_{k=1}^{n} |V_i V_k Y_{ik}| \cos(\theta_{ik}+\delta_k-\delta_i) \quad i=1,2,\ldots n \tag{3.14}$$

$$Q_i = QG_i - QD_i = -\sum_{k=1}^{n} |V_i V_k Y_{ik}| \sin(\theta_{ik}+\delta_k-\delta_i) \quad i=1,2,\ldots n \tag{3.15}$$

An examination of Equations (3.14) and (3.15) shows that only four of the eight variables PG, QG, PD, QD, V, δ, Y, and θ are known. That is, Y and θ are specified by the transmission network parameters and PD and QD are the known load requirements imposed on the system. Consequently, a method must be devised to reduce the number of unknowns to two per bus since there are only two independent equations in (3.14) and (3.15) per bus. The problem is resolved by classifying the buses into three types according to the function they serve in the system. The three classifications are: type 1--a load bus, type 2--a voltage control bus, and type 3 is the slack bus. These types are discussed below and allow two of the four unknowns at each bus to be fixed or specified according to the type bus.

The slack bus, as indicated earlier, is used as the voltage reference for the system. Therefore, there is only one slack bus per system and the voltage magnitude $|V_i|$ and phase angle δ_i at this bus can be set arbitrarily, usually at a per unit value of 1.0 and 0.0 respectively. Although the load requirements PD_i+jQD_i are known throughout the system, the total generated power requirements can not be specified because the total system losses (TPL) are unknown. That is, the total generated power must equal the sum of the various loads plus the total system losses.

$$\sum_{i=1}^{n} (PG_i + jQG_i) = \sum_{i=1}^{n} (PD_i + jQD_i) + TPL \quad i=1,2,\ldots n \tag{3.16}$$

Consequently, the real and reactive powers are the unknowns at the slack bus so that the extra power needed to compensate for the system losses can be included in the solution for generated power at this bus.

At other buses it is often desirable to control the voltage magnitude at a specific value for better control and service throughout the system. At these "voltage control" buses the generated real power PG_i and the voltage magnitude $|V_i|$ are specified while the phase angle δ_i and the required reactive power QG_i are the unknowns. In this case, the necessary reactive power to maintain the specified bus voltage may be supplied by a generator or a compensating capacitor bank connected to the bus as shown on bus 3 in Figure 3.2. Inductor banks are also used to consume excess reactive power such as often occurs on long distance very high voltage transmission lines. At the remaining "load" type buses, the generated real and reactive power is specified and the voltage magnitude and phase angles are the unknowns.

Solution of the load flow problem is now possible since with an n bus system there will be 2n unknowns and 2n real equations given in (3.14) and (3.15). Table 3.3 illustrates this by showing the "Bus Data" in per unit values for the 4-bus system shown in Figure 3.2. In the EDSP code, a negative number is used for the bus type to indicate those buses with generators attached. As shown, there are now 8 unknowns in the system of 8 independent real load flow equations. They are:

$$\begin{aligned}
1.25-0.9 &= |V_1{}^2Y_{11}|\cos(\theta_{11}) \\
&+ |V_1Y_{12}|\cos(\theta_{12}-\delta_1) + |V_1Y_{13}|\cos(\theta_{13}+\delta_3-\delta_1) \\
PG_2-0.5 &= |V_1Y_{21}|\cos(\theta_{21}+\delta_1) \\
&+ |Y_{22}|\cos(\theta_{22}) + |V_4Y_{24}|\cos(\theta_{24}+\delta_4) \\
0.0-0.4 &= |V_1Y_{31}|\cos(\theta_{31}+\delta_1-\delta_3) \\
&+ |Y_{33}|\cos(\theta_{33}) + |V_4Y_{34}|\cos(\theta_{34}+\delta_4-\delta_3) \\
0.0-0.6 &= |V_4Y_{42}|\cos(\theta_{42}-\delta_4) \\
&+ |V_4Y_{43}|\cos(\theta_{43}+\delta_3-\delta_4) + |V_4{}^2Y_{44}|\cos(\theta_{44})
\end{aligned} \tag{3.17}$$

Table 3.3 Bus Data

Bus No.	Bus Type	PG	QG	PD	QD	V	S
1	-1	1.25	0.8	0.9	0.4	?	?
2	-3	?	?	0.5	0.4	1.0	0.0
3	2	0.0	?	0.4	0.4	1.0	?
4	1	0.0	0.0	0.6	0.3	?	?

$$\begin{aligned}
0.8-0.4 &= -|V_1{}^2Y_{11}|\sin(\theta_{11}) \\
&\quad - |V_1Y_{12}|\sin(\theta_{12}-\delta_1) - |V_1Y_{13}|\sin(\theta_{13}+\delta_3-\delta_1) \\
QG_2-0.4 &= -|V_1Y_{21}|\sin(\theta_{21}+\delta_i) \\
&\quad - |Y_{22}|\sin(\theta_{22}) - |V_4Y_{24}|\sin(\theta_{24}+\delta_4) \\
QG_3-0.4 &= -\ |V_1Y_{31}|\sin(\theta_{31}+\delta_1-\delta_3) \\
&\quad - |Y_{33}|\sin(\theta_{33}) - |V_4Y_{34}|\sin(\theta_{34}+\delta_4-\delta_3) \\
0.0-0.3 &= -\ |V_4Y_{42}|\sin(\theta_{42}-\delta_4) \\
&\quad - |V_4Y_{43}|\sin(\theta_{43}+\delta_3-\delta_4) - |V_4{}^2Y_{44}|\sin(\theta_{44})
\end{aligned} \tag{3.18}$$

Note that the $|Y|$s and θs are known quantities and the remaining 8 unknowns are: $|V_1|$, δ_1, PG_2, QG_2, QG_3, δ_3, $|V_4|$, and δ_4.

The Economic Dispatch Problem

Today fuel costs exceed 87 per cent of the total plant operating expense (see, e.g., Friedlander, 1977:48). Thus, for the solution of the economic dispatch problem (i.e., to determine the minimum cost power generation profile that satisfies the total system load), a relationship between the cost of fuel input and electrical power output is required. The plant input-output curves are used for this purpose. The curves define the heat input required to produce the net electrical power output, usually expressed in BTU per Kilowatt hour (see, e.g., Skrotzki and Vopat, 1960:612-617). These curves are not linear and are determined empirically. For steam power plants, they are usually obtained by steady state operation of the generator prime mover at "valve points," point just prior to the next steam control valve opening, from generator no-load up to the generator peak rated output. The fuel required at each valve point is multiplied by the fuel cost to obtain the generator cost curves. Starting fuel and boiler banking fuel is not included since these fuel expenditures are independent of system loads. Thus, the cost function C_i used for generator i is defined as the cost in dollars per hour to produce its net electrical power output S_{Gi}, e.g., see Figure 3.3.

$$C_i \equiv C_i(PG_i) \quad \$/Hr \qquad (3.19)$$

The cost function C_i is assumed to be a function of the real power output PG_i only since the real power is controlled by the prime mover torque (Elgerd, 1971:276). However, the generated reactive power QG_i does have an indirect influence on operating cost. That is, the reactive power is controlled by varying the generator field current thereby varying the bus voltages throughout the system. Since the transmission losses are a function of these bus voltage levels, the reactive power implicitly influences the system operating cost through its effect on system losses and their demand for increased real power output. It is recognized that the use of these "smooth" cost curves ignores the detrimental effects of operating the steam turbines between "valve points" where steam throttling losses are incurred. However, studies by Ringlee and Williams (1962) and Happ, Ille, and Reisinger (1962) indicate that a total system cost savings of only 0.1 to 0.2 per cent could be obtained by use of dispatching criteria that includes these valve throttling effects. Therefore, incorporation of these refinements was left for further study when actual operating and cost data become available for comparison.

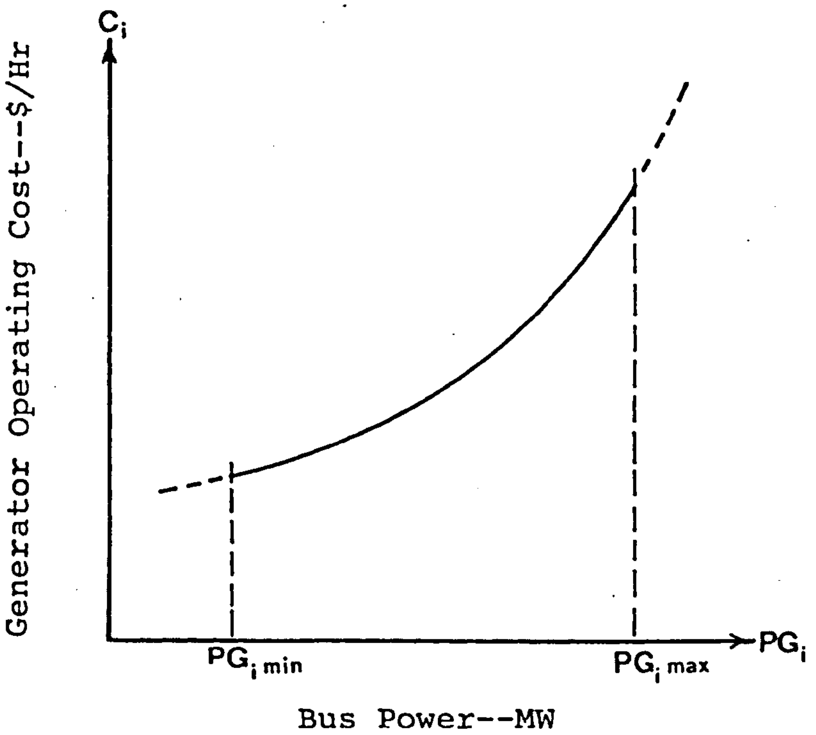


Figure 3.3 Typical Generator Cost Curve

The first derivative of the cost curve C_i is defined as the incremental generator operating cost IC_i in dollars per megawatt hour. The incremental cost curves represent the slope of the cost curve at any point and can be obtained by graphical differentiation or curve fitting techniques. They are then approximated by the following second order polynomial:

$$IC_i \equiv \frac{\partial C_i}{\partial PG_i} = \alpha_i + \beta_i(PG_i) + \gamma_i(PG_i)^2 \quad \$/\text{MWHr} \qquad (3.20)$$

The EDSP computer code stores the incremental cost curve coefficients α_i, β_i, γ_i, normalized in per unit values for use in developing an equal incremental cost generation schedule for all generators. The normalized form is obtained by,

$$IC_i \text{ p.u.} = \frac{IC_i \times S_b}{C_b} \qquad (3.21)$$

where S_b and C_b are the arbitrarily chosen system power base in MVA and the system operating cost base in dollars per hour, respectively. Figure 3.4 shows the cost and incremental cost curves in per unit values used for the 4-bus system in Figure 3.2 and Table 3.4 shows the cost curve coefficients and generator constraints used in per unit values.

Equipment such as generators, capacitor banks, transformers, etc., have operating limits beyond which

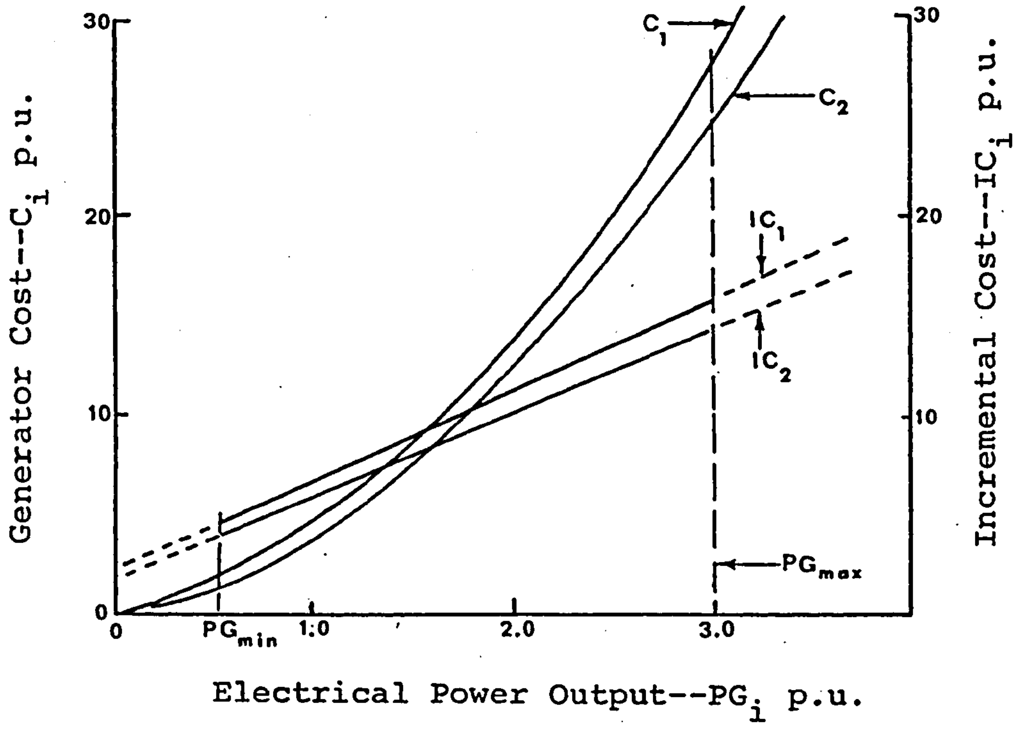


Figure 3.4 Cost and Incremental Cost Curves for Typical 4-Bus System

Table 3.4 Generator Data and Constraints

Bus No.	Alpha	Beta	Gamma
1	2.4	4.5	0.01
2	2.0	4.1	0.01

Generator Constraints

Bus No.	PG_imax	PG_imin	QG_imax	QG_imin
1	3.0	0.5	--	--
2	3.0	0.5	--	--
3	--	--	2.0	0.0

operation of the equipment will result in reduced service live, unpredictable operation, or possible complete destruction of the equipment. The code accepts these operating limits as "Generator Constraints" during problem solution. When a limit is reached the variable is held at the limit value and the solution continued with remaining unconstrained generators sharing the remaining load. The following variables are kept within the limits indicated:

$$PG_i\text{min} \leq PG_i \leq PG_i\text{max} \tag{3.22}$$

$$QG_i\text{min} \leq QG_i \leq QG_i\text{max} \tag{3.23}$$

The EDSP computer code treats capacitor banks used for reactive power compensation to control the bus voltage magnitude at type 2 buses as generators without real power output capability. Also, when the capacitor bank or generator reactive power QG_i reaches its limit in an attempt to control the voltage at its specified level, the reactive power is held at its limit value and the bus is changed to a type 1 bus. From this point on the voltage is varied until a solution is obtained. During economic dispatch solutions where successive load flow solutions are performed, as discussed in Chapter 5, the bus type is returned to a type 2 between load flow solution iterations to permit "backoff" from the limits during power excursions in the economic dispatch calculations. An example of the input generator incremental cost data and constraint required by the computer code is shown in Table 3.4 for the 4-bus system in Figure 3.2. All entries are in per unit values. Note that reactive power constraints are required for type 2 voltage control buses only.

With the incremental cost functions and operating constraints of each generator defined, the economic dispatch or optimum load flow problem is to develop a process whereby the total power generated within the system is allocated among the available generating units in such a way as to minimize the total system operating cost.

An iterative solution method is indicated since an infinite number of generator settings could satisfy the system load demands. Also, it is clear that the three functions: the generator cost curves (3.19) and (3.20), the system power balance equation (3.16), and the equipment constraints in equations (3.22) and (3.25) must be related in such a way that an optimum solution can be found.

The nonlinear programming theorem developed by Kuhn and Tucker (1951) and first applied to the economic dispatch problem by Carpentier (1962a, 1962b) provides the method for obtaining the optimum cost generation profile. To apply this theorem to the economic dispatch problem, we first define the total system operating cost as the sum of the individual cost functions.

$$C \equiv \sum_{i=1}^{n} C_i(PG_i) \quad i = 1,2,\ldots n \tag{3.24}$$

Then using the power balance equation (3.16), we define the function $h(PG_i)$ as the system real power balance,

$$h(PG_1,PG_2,\ldots PG_n) \equiv \sum_{i=1}^{n} (PG_i - PD_i) - TPL = 0 \qquad i=1,2,\ldots n \tag{3.25}$$

Now Equations (3.24) and (3.25) are combined to form an augmented cost function defined as,

$$C^* \equiv \sum_{i=1}^{n} C_i - \lambda(\sum_{i=1}^{n} PG_i - \sum_{i=1}^{n} PD_i - TPL) \quad i=1,2,\ldots n \tag{3.26}$$

The partial derivative of C* with respect to PG provides the optimum dispatch equation,

$$\frac{\partial C^*}{\partial PG_i} = \frac{\partial \sum_{i=1}^{n} C_i}{\partial PG_i} - \lambda (\frac{\partial \sum_{i=1}^{n} (PG_i - PD_i)}{\partial PG_i} - \frac{\partial TPL}{\partial PG_i}) = 0 \tag{3.27}$$

In this equation the partial derivative of the total transmission losses TPL with respect to PG_i is defined as the "Incremental Transmission Loss" ITL_i associated with the generating unit i. Elgerd (1971:296-299) has developed the following good approximation of ITL in terms of the variables already obtained by the load flow solution:

$$\frac{\partial PL}{\partial G_i} \equiv ITL_i \equiv 2 \sum_{k=1}^{n} (P_k\alpha_{ik} - Q_k\beta_{ik}) \quad i=1,2,\ldots n \tag{3.28}$$

where the terms α_{ik} and β_{ik} are defined as follows:

$$\alpha_{ik} \equiv \frac{r_{ik}}{|V_i V_k|} \cos(\delta_i - \delta_k) \tag{3.29}$$

$$\beta_{ik} \equiv \frac{r_{ik}}{|V_i V_k|} \sin(\delta_i - \delta_k) \tag{3.30}$$

and r_{ik} is the real part of the inverse Y admittance matrix element.

$$r_{ik} \equiv \text{Re:}\ (|Y_{ik}| \underline{/\theta_{ik}})^{-1} \tag{3.31}$$

When the differentiation indicated by (3.27) is performed, an expression is obtained for the incremental operating cost of each generating unit in terms of the transmission losses associated with the distribution of its power output.

$$IC_i = \lambda(1-ITL_i) \quad i=1,2,\ldots n \tag{3.32}$$

Thus, as shown by Elgerd (1971:294), the concept of scheduling power production by equal incremental cost of each generating unit can now be modified to account for the losses incurred by dispatching it through the transmission network to the load. Figure 3.5 gives a graphic representation of how Equation (3.32) obtains the optimum incremental cost profile using the family of individual generator incremental cost curves. It can be seen that an iterative process where the value of λ is varied until the power balance equation (3.25) is satisfied will produce the optimum incremental cost schedule for each generator.

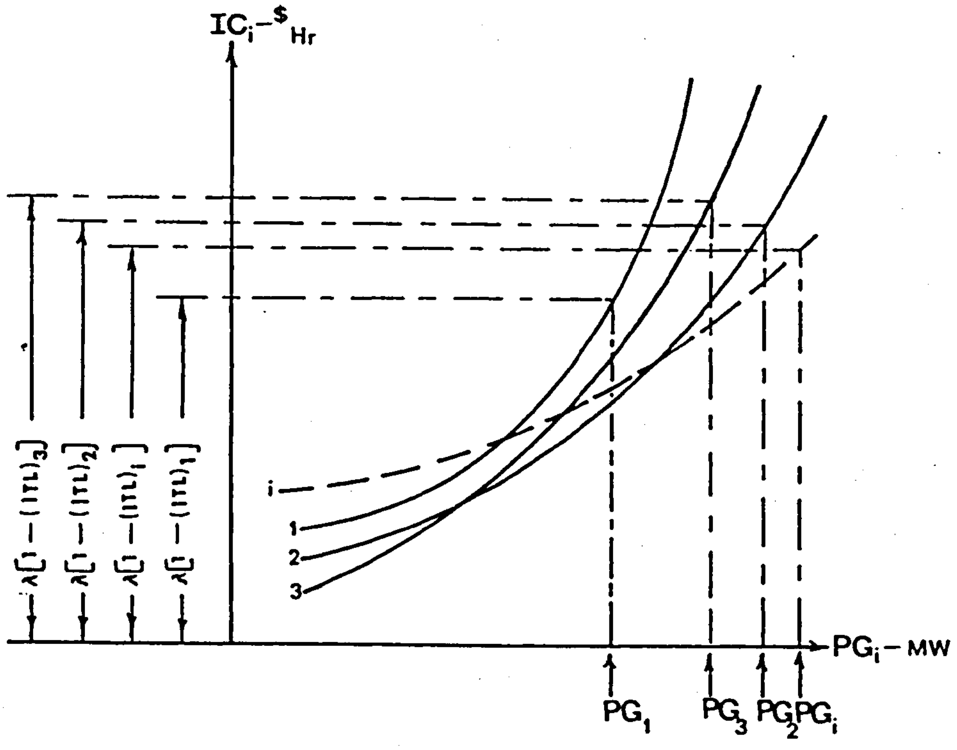


Figure 3.5 Graphical Economic Dispatch Solution

CHAPTER 4

LOAD FLOW ANALYSIS

The objective of the load flow study is to determine the bus voltages and phase angles associated with a selected schedule of power generation that meets the given set of load demands. With the voltages known, the line flows and system power losses can be calculated directly. Since the losses are not known, a priori, an iterative solution method with the nonlinear load flow equations (3.14) and (3.15) is indicated. In this chapter the iterative Newton-Raphson solution method used will be developed to assist the system engineer in the analysis of problem areas in power system load flow studies.

The main difficulty with successive displacement or Gauss-Seidel solution methods are their inability to solve some types of problems, such as those involving negative reactances that may arise in the equivalent circuits for 3-winding transformers or problems in which high and low impedance branches terminate at the same bus. The Newton-Raphson solution method was selected for use in the computer code because, as outlined by Tinney and Hart (1967:1450), it is not affected by these and most other ill-conditioned situations. Also, the Newton-Raphson method provides a more rapid convergence rate, usually not more than 3 to 5 iterations, and the number of iterations is relatively independent of the problem size. However, this method is extremely sensitive to the initial starting conditions (see, e.g., Gross and Luini, 1971:41-48). Fortunately, in power system problems, the use of a flat voltage profile, $V_i\underline{/\delta_i} = 1.0\underline{/0.0}$ per unit, as the initial starting point is sufficiently near the final solution point to preclude a false solution or divergence.

The iterative Newton-Raphson solution method used is summarized as follows:

Given the set of n nonlinear equations,

$$f_i(x_1,x_2...x_n) = 0 \qquad i=1,2,...n \tag{4.1}$$

and let,

$$\bar{f} \equiv \begin{bmatrix} f_1 \\ \cdot \\ \cdot \\ \cdot \\ f_n \end{bmatrix}, \; \bar{x} \equiv \begin{bmatrix} x_1 \\ \cdot \\ \cdot \\ \cdot \\ x_n \end{bmatrix}, \; \overline{\Delta x} \equiv \begin{bmatrix} \Delta x_1 \\ \cdot \\ \cdot \\ \cdot \\ \Delta x_n \end{bmatrix}$$

In order to solve this set of n nonlinear simultaneous equations we make an initial estimate of $\bar{x}$. Call the estimated value $\bar{x}^{(o)}$, then let $\Delta x_1{}^{(o)},...\Delta x_n{}^{(o)}$ be the difference between $x_i{}^{(o)}$ and the exact value x_i, i.e., $x_i{}^{(o)} + \Delta x_i{}^{(o)} = x_i$. If the estimated value $x_i{}^{(o)}$ is close to the exact value x_i, then the differential df_i is:

$$df_i = \sum_{j=1}^{n} \left|\frac{\partial f_i}{\partial x_j}\right|_{\bar{x}^{(o)}} \overline{\Delta x}_j^{(o)} = f_i(\bar{x}) - f_i(\bar{x}^{(o)}) \quad i=1,2,\ldots n \tag{4.2}$$

and if,

$$d\bar{f} \equiv \begin{bmatrix} f_1(\bar{x}) - f_1(\bar{x}^{(o)}) \\ \vdots \qquad \vdots \\ f_n(\bar{x}) - f_n(\bar{x}^{(o)}) \end{bmatrix} ,$$

then

$$d\bar{f} = [J](\overline{\Delta x}^{(o)}) = \bar{f}(\bar{x}) - \bar{f}(\bar{x}^{(o)})$$

where [J] is a matrix with elements $J_{ij} = (\frac{\partial f_i}{\partial x_j})_{\bar{x}(o)}$ i,j=1,2,...n. Solving for $\overline{\Delta x}^{(o)}$ and recognizing from (4.1) that $\bar{f}(\bar{x}) = 0$, we obtain the general Newton-Raphson form:

$$\overline{\Delta x}^{(o)} = -[J]^{-1}\bar{f}(\bar{x}^{(o)}) \tag{4.3}$$

Thus, we may determine $\bar{x}$ approximately by,

$$\bar{x}^{(1)} = \bar{x}^{(o)} + \overline{\Delta x}^{(o)}$$

where $\bar{x}^{(1)}$ may not be exactly equal to $\bar{x}$ since the Taylor expansion of $d\bar{f}$ in (4.2) only retains the first order terms. However, $\bar{x}^{(1)}$ is a better approximation of $\bar{x}$ than the initial estimate $\bar{x}^{(o)}$, and so the process is repeated to obtain an even better approximation, $\bar{x}^{(2)}$. The iterative procedure is continued until $\overline{\Delta x}^{(p)}$ is equal to or less than some acceptable solution tolerance. In the EDSP computer codes, this solution tolerance (TOL) is preset at 0.0001, but can be changed by the user during program execution.

To apply Newton's method to the load flow problem, let two functions $g_i(P,V,\delta)$ and $h_i(Q,V,\delta)$ be defined so that they approach zero as the estimated values of real and reactive power approach the exact values.

$$g_i(P,V,\delta) = P_i^{\text{exact}} - P_i^{\text{estimated}} \longrightarrow 0 \qquad (4.4)$$

$$h_i(Q,V,\delta) = Q_i^{\text{exact}} - Q_i^{\text{estimated}} \longrightarrow 0 \qquad (4.5)$$

If the load flow equations (3.14) and (3.15) are substituted into (4.4) and (4.5) for the estimated values of P_i and Q_i, expressions in terms of the four variables P, Q, V, δ, are obtained.

$$g_i = P_i^{\text{exact}} - \sum_{k=1}^{n} |V_i V_j Y_{ik}| \cos(\theta_{ik}+\delta_k-\delta_i) \qquad i=1,2,\ldots n \qquad (4.6)$$

$$h_i = Q_i^{\text{exact}} + \sum_{k=1}^{n} |V_i V_k Y_{ik}| \sin(\theta_{ik}+\delta_k-\delta_i) \qquad i=1,2,\ldots n \qquad (4.7)$$

Now if the estimated value g_i^P is close to the final solution, the differential dg_i is:

$$dg_i = g_1^{\text{exact}} - g_i^P$$

$$= \frac{\partial g_i^P}{\partial P_i} dP_i + \sum_{k=1}^{n} \frac{\partial g_i^P}{\partial \delta_k} d\delta_k + \sum_{k=1}^{n} \frac{\partial g_i^P}{\partial V_k} dV_k \qquad (4.8)$$

$i=1,2,\ldots n$ n=number of buses

However, from (4.4) g_i^{exact} is equal to zero, therefore the estimated value g_i^p is equal to,

$$g_i^p = - \frac{\partial g_i^p}{\partial P_i} dP_i - \sum_{k=1}^{n} \frac{\partial g_i^p}{\partial \delta_k} d\delta_k - \sum_{k=1}^{n} \frac{\partial g_i^p}{\partial V_k} dV_k \tag{4.9}$$

and similarly,

$$h_i^p = - \frac{\partial h_i^p}{\partial P_i} dP_i - \sum_{k=1}^{n} \frac{\partial h_i^p}{\partial \delta_k} d\delta_k - \sum_{k=1}^{n} \frac{\partial h_i^p}{\partial V_k} dV_k \tag{4.10}$$

Thus, in Newton-Raphson form the solution is:

$$(\Delta\bar{g}_i^p) = - [J(\bar{g}_i^p)]^{-1} f(\bar{g}_i^p) \tag{4.11}$$

and

$$(\Delta\bar{h}_i^p) = - [J(\bar{h}_i^p)]^{-1} f(\bar{h}_i^p) \tag{4.12}$$

Where [J] is the jacobian matrix of $f(\bar{g}_i^p)$ and $f(\bar{h}_i^p)$ respectively. The next estimate of the independent variables, P, Q, V, and δ is obtained by solving (4.11) and (4.12) for Δg_i^p and Δh_i^p respectively and adding them to the original estimated values,

$$g_i^{p+1} = g_i^p + \Delta g_i^p \tag{4.13}$$

and

$$h_i^{p+1} = h_i^p + \Delta h_i^p \tag{4.14}$$

The solution continues until the changes in P, Q, V, δ are less than or equal to the acceptable solution tolerance, i.e., TOL.

In Chapter 3 it was established that the unknown variables depend on the bus type. Consequently, the solution matrix will be modified based on the type buses in the system. That is, the elements of the jacobian will be 0, -1, or the partial derivative of the function g_i or h_i depending on the bus type.

Consider the following example based on the 4-bus system shown in Figure 3.2. The bus types and corresponding unknown variables are shown in Table 4.1. In the solution method, the unknown variables V_1, δ_1, P_2, Q_2, Q_3, δ_3, and V_4 are given initial estimated values and then Equations (4.11) and (4.12) are solved for ΔP, ΔQ, ΔV, $\Delta\delta$. The first iteration is shown in (4.15) and is solved for the changes in P, Q, V, δ.

Table 4.1 Bus Characteristics

Bus type	PG_i	QG_i	V_i	δ_i
Load (Type 1)	Specified	Specified	?	?
Control (Type 2)	Specified	?	Specified	?
Slack (Type 3)	?	?	Specified	Specified

$$\left[\begin{array}{c} g_1 \\ g_2 \\ g_3 \\ g_4 \\ \hline h_1 \\ h_2 \\ h_3 \\ h_4 \end{array}\right] = - \left[\begin{array}{cccc:cccc} \frac{\partial g_1}{\partial \delta_1} & 0 & \frac{\partial g_1}{\partial \delta_3} & \frac{\partial g_1}{\partial \delta_4} & \frac{\partial g_1}{\partial V_1} & 0 & 0 & \frac{\partial g_1}{\partial V_4} \\ \frac{\partial g_2}{\partial \delta_1} & -1 & \frac{\partial g_2}{\partial \delta_3} & \frac{\partial g_2}{\partial \delta_4} & \frac{\partial g_2}{\partial V_1} & 0 & 0 & \frac{\partial g_2}{\partial V_4} \\ \frac{\partial g_3}{\partial \delta_1} & 0 & \frac{\partial g_3}{\partial \delta_3} & \frac{\partial g_3}{\partial \delta_3} & \frac{\partial g_3}{\partial V_1} & 0 & 0 & \frac{\partial g_3}{\partial V_4} \\ \frac{\partial g_4}{\partial \delta_1} & 0 & \frac{\partial g_4}{\partial \delta_3} & \frac{\partial g_4}{\partial \delta_4} & \frac{\partial g_4}{\partial V_1} & 0 & 0 & \frac{\partial g_4}{\partial V_4} \\ \hdashline \frac{\partial h_1}{\partial \delta_1} & 0 & \frac{\partial h_1}{\partial \delta_3} & \frac{\partial h_1}{\partial \delta_4} & \frac{\partial h_1}{\partial V_1} & 0 & 0 & \frac{\partial h_1}{\partial V_4} \\ \frac{\partial h_2}{\partial \delta_1} & 0 & \frac{\partial h_2}{\partial \delta_3} & \frac{\partial h_2}{\partial \delta_3} & \frac{\partial h_2}{\partial V_1} & -1 & 0 & \frac{\partial h_2}{\partial V_4} \\ \frac{\partial h_3}{\partial \delta_1} & 0 & \frac{\partial h_3}{\partial \delta_3} & \frac{\partial h_3}{\partial \delta_4} & \frac{\partial h_3}{\partial V_1} & 0 & -1 & \frac{\partial h_3}{\partial V_4} \\ \frac{\partial h_4}{\partial \delta_1} & 0 & \frac{\partial h_4}{\partial \delta_3} & \frac{\partial h_4}{\partial \delta_3} & \frac{\partial h_4}{\partial V_1} & 0 & 0 & \frac{\partial h_4}{\partial V_4} \end{array}\right] \left[\begin{array}{c} \Delta\delta_1 \\ \Delta P_2 \\ \Delta\delta_3 \\ \Delta\delta_4 \\ \hline \Delta V_1 \\ \Delta Q_2 \\ \Delta Q_3 \\ \Delta V_4 \end{array}\right] \tag{4.15}$$

To solve for the new estimated values of P, Q, V, and δ, the jacobian matrix must be inverted. That is,

$$\bar{g}_i^{\,p+1} = \bar{g}_i^{\,p} - [J(g_i^{\,p})]^{-1} \bar{f}(\bar{g}_i^{\,p})$$

$$\bar{h}_i^{\,p+1} = \bar{h}_i^{\,p} - [J(h_i^{\,p})]^{-1} \bar{f}(\bar{h}_i^{\,p}) \tag{4.17}$$

The jacobian matrix is partitioned with the elements of the partitioned sub-matrices defined as follows:

$$[J] = \left[\begin{array}{c|c} A_1 & A_2 \\ \hline A_3 & A_4 \end{array}\right] \qquad (4.18)$$

$$[A_1] \equiv \left[\begin{array}{l} \underline{\text{On Diagonal Elements } (i=k)} \\ \quad \text{Bus Types 1 and 2} \\ \quad J_{ii} = \sum_{\substack{k=1 \\ k\neq i}}^{n} |V_i V_k Y_{ik}| \sin(\theta_{ik}+\delta_k-\delta_i) \quad i=1,2,\ldots n \\ \quad \text{Bus Type 3} \\ \quad J_{ii} = -1 \\ \underline{\text{Off Diagonal Elements } (i\neq k)} \\ \quad \text{Bus Types 1 and 2} \\ \quad J_{ik} = -|V_i V_k Y_{ik}| \sin(\theta_{ik}+\delta_k-\delta_i) \quad i=1,2,\ldots n \\ \quad \text{Bus Type 3} \\ \quad J_{ik} = 0 \end{array}\right]$$

(4.19)

$$[A_2] \equiv \left[\begin{array}{l}
\underline{\text{On Diagonal Elements } (i=k)} \\
\quad \text{Bus Type 1} \\
\quad J_{ii} = 2|V_i Y_{ik}|\cos(\theta_{ii}) \\
\quad + \sum_{\substack{k=1 \\ k\neq i}}^{n} |V_k Y_{ik}|\cos(\theta_{ik}+\delta_k-\delta_i) \quad i=1,2,\ldots n \\
\quad \text{Bus Types 2 and 3} \\
\quad J_{ii}=0 \\
\\
\underline{\text{Off Diagonal Elements } (i\neq k)} \\
\quad \text{Bus Type 1} \\
\quad J_{ik} = |V_i Y_{ik}|\cos(\theta_{ik}+\delta_k-\delta_i) \\
\quad \text{Bus Types 2 and 3} \\
\quad J_{ik} = 0
\end{array}\right] \tag{4.20}$$

$$[A_3] \equiv \left[\begin{array}{l} \underline{\text{On Diagonal Elements } (i=k)} \\ \quad \text{Bus Types 1 and 2} \\ \quad J_{ii} = \sum_{\substack{k=1\\ k\neq i}}^{n} |V_i V_k Y_{ik}| \cos(\theta_{ik}+\delta_k-\delta_i) \quad i=1,2,\ldots n \\ \quad \text{Bus Type 3} \\ \quad J_{ii} = 0 \\ \\ \underline{\text{Off Diagonal Elements } (i\neq k)} \\ \quad \text{Bus Types 1 and 2} \\ \quad J_{ik} = -|V_i V_k Y_{ik}| \cos(\theta_{ik}+\delta_k-\delta_i) \quad i=1,2,\ldots n \\ \quad \text{Bus Type 3} \\ \quad J_{ik} = 0 \end{array}\right] \tag{4.21}$$

$$[A_4] \equiv \left[\begin{array}{l}
\underline{\text{On Diagonal Elements } (i=k)} \\
\quad \text{Bus Type 1} \\
\quad J_{ii} = -2|V_iY_{ii}|\sin(\theta_{ii}) \\
\quad -\sum\limits_{\substack{k=1 \\ k\neq i}}^{n} |V_kY_{ik}|\sin(\theta_{ik}+\delta_k-\delta_i) \quad i=1,2,\ldots n \\
\quad \text{Bus Types 2 and 3} \\
\quad J_{ii} = -1 \\
\\
\underline{\text{Off Diagonal Elements } (i\neq k)} \\
\quad \text{Bus Type 1} \\
\quad J_{ik} = -|V_iY_{ik}|\sin(\theta_{ik}+\delta_k-\delta_i) \\
\quad \text{Bus Types 2 and 3} \\
\quad J_{ik} = 0
\end{array}\right] \tag{4.22}$$

The inverse of the partitioned jacobian is obtained as follows:

$$[J]^{-1} = \left[\begin{array}{c:c} B_1 & B_2 \\ \hdashline B_3 & B_4 \end{array}\right] \tag{4.23}$$

where the partitioned sub-matrices are obtained using (4.19) to (4.22) as outlined by Stagg and El-Abiad (1968: 19).

$$B_1 = (A_1 - A_2 A_4^{-1} A_3)^{-1}$$

$$B_2 = -B_1 A_2 A_4^{-1}$$

$$B_3 = -A_4^{-1} A_3 B_1$$

$$B_4 = A_4^{-1} - A_4^{-1} A_3 \; B_2$$

This allows Equation (4.15) to be solved for the right hand column vector and then with (4.16) and (4.17) obtain the first order approximation of the solution. Further iterations will improve the approximations until the solution is obtained within the desired accuracy tolerance.

Solution Method and Flow Chart

The overall strategy of the solution method can be summarized as follows:

1. All system parameters are input, generator constraints established, and convergence criteria specified. These include the transmission line data, bus data, generator data and limits, solution tolerance and maximum number of iterations desired, and the initial estimated values for the unknowns.
2. The system transmission network admittance matrix [Y] is calculated. This calculation need be performed only once for each network configuration. The results are stored in the temporary SOS file

FOR24.DAT for subsequent studies using the same distribution network.

3. Using the initial voltage and phase angle estimates at each bus, calculate the resulting net real and reactive power at each bus using the load flow equations (3.14) and (3.15).

4. Test for convergence. The specified values of real and reactive power are compared with the values calculated in 3 above using Equations (4.4) and (4.5). When the difference becomes less than the desired solution tolerance "TOL" the final solution has been attained and the power balance equation (3.16) has been satisfied.

5. If the desired tolerance is exceeded at any bus, the solution continues using Equations (4.16) and (4.17) to determine the new estimated values of the unknown variables P, Q, V, and δ. The new estimated values of reactive power at type 2 buses are then checked against the limits designated. If any limit is exceeded the limit values are substituted and the solution continued changing the type 2 bus into a type 1 bus so that convergence can be attained.

6. Steps 3 through 5 are repeated using the new estimated values of P, Q, V, and δ after each iteration until the solution is obtained or the maximum number of iterations has been reached.

7. When a final solution is reached, the power flow in each transmission line is calculated where,

$$P_{ik} = |V_i^2 Y_{ik}|\cos(\theta_{ik}) - |V_i V_k Y_{ik}|\cos(\delta_i - \delta_k - \theta_{ik})$$

$$+ |V_i V_k Y_{shi}|\cos(\theta_{shi}) \qquad (4.24)$$

$$Q_{ik} = -|V_i^2 Y_{ik}|\sin(\theta_{ik}) - |V_i V_k Y_{ik}|\sin(\delta_i - \delta_k - \theta_{ik})$$

$$+ |V_i V_k Y_{shi}|\sin(\theta_{shi}) \qquad (4.25)$$

8. Calculate the total system load and transmission losses as follows:

$$\text{Total system load TPD} = \sum_{i=1}^{n} PD_i \quad i=1,2,\ldots n \qquad (4.26)$$

$$\text{Total system losses TPL} = \sum_{i=1}^{n} P_i \quad i=1,2,\ldots n \qquad (4.27)$$

9. Print desired output.

Sample problems are included in Appendix A showing input/output options and format. Figure 4.1 contains a flow chart showing the principal steps and sequence of each step in the load flow solution.

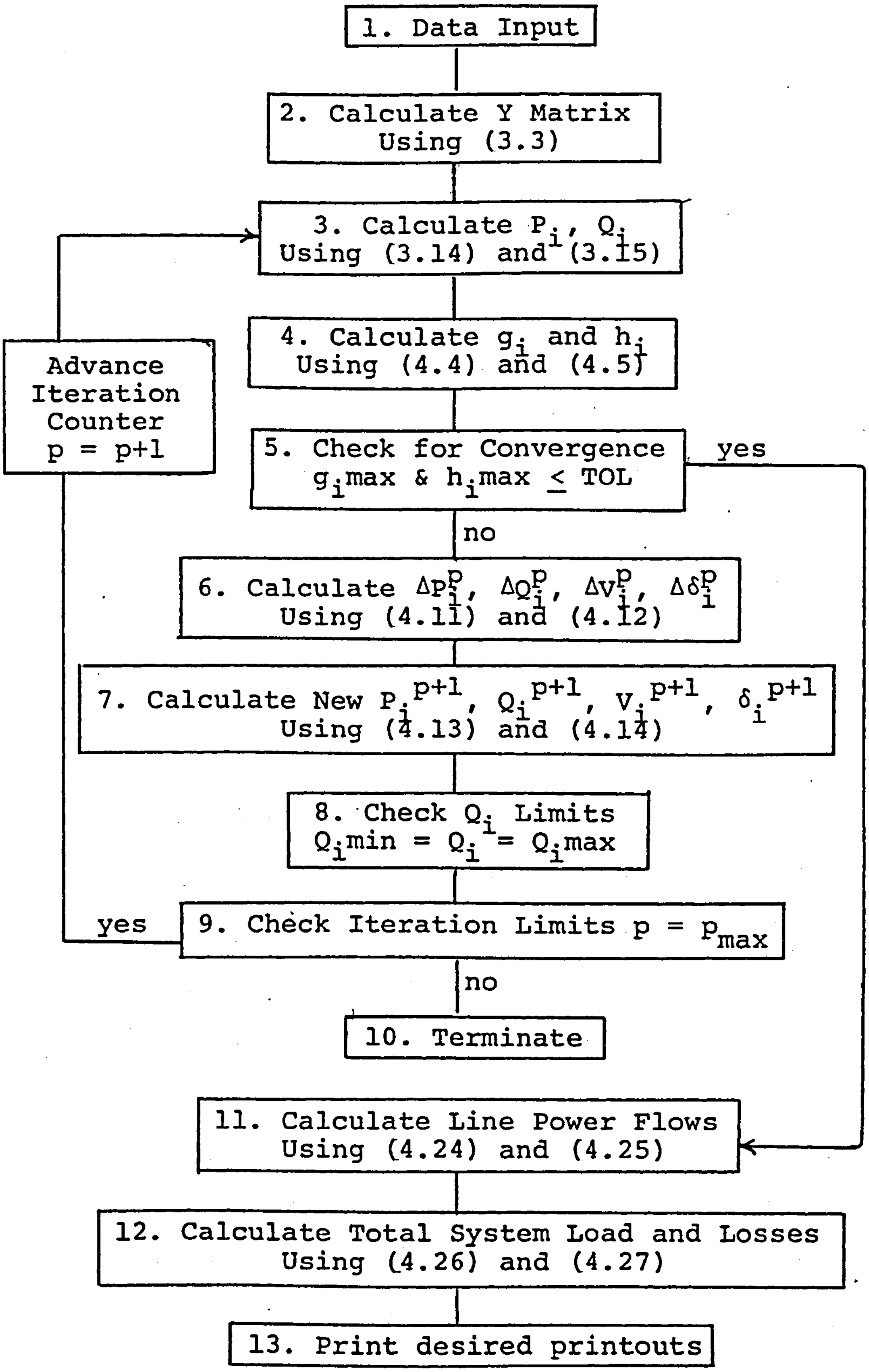

1. Data Input
2. Calculate Y Matrix Using (3.3)
3. Calculate P_i, Q_i Using (3.14) and (3.15)
4. Calculate g_i and h_i Using (4.4) and (4.5)
5. Check for Convergence $g_i$max & $h_i$max $\leq$ TOL
yes
no
6. Calculate ΔP_i^p, ΔQ_i^p, ΔV_i^p, $\Delta \delta_i^p$ Using (4.11) and (4.12)
7. Calculate New P_i^{p+1}, Q_i^{p+1}, V_i^{p+1}, δ_i^{p+1} Using (4.13) and (4.14)
8. Check Q_i Limits $Q_i$min = Q_i = $Q_i$max
9. Check Iteration Limits $p = p_{max}$
yes
Advance Iteration Counter $p = p+1$
no
10. Terminate
11. Calculate Line Power Flows Using (4.24) and (4.25)
12. Calculate Total System Load and Losses Using (4.26) and (4.27)
13. Print desired printouts

Figure 4.1 Load Flow Solution Flow Chart

CHAPTER 5

ECONOMIC DISPATCH ANALYSIS

In the previous chapter a method was developed for determining the power distribution throughout the system when the system engineer specified two of the four variables at each bus, i.e., Table 4.1. In this load flow solution, the specified variables were selected arbitrarily by the system engineer as an "educated guess." To obtain the optimum or most economical power generation profile, the total system power generation requirement, as calculated by the load flow solution, is re-scheduled among the available generating units using the incremental production cost characteristics of each generator in such a way as to minimize the total system operating cost in dollars per hour \$/Hr. In Chapter 3 it was shown that the optimum generation profile is obtained by an iterative process varying the Lagrangian multiplier λ in Equation (3.32) and using the incremental cost IC_i of each generator in Equation (3.20) to obtain the real power generated PG_i by each generating unit. λ is varied until the sum of the individual power generation PG_i satisfies the total system requirement, Equation (3.25), within an acceptable tolerance ESP1. This process is constrained by the operating limits of each generator. That is, if a generator output limit is reached, its output is fixed at the limit value and the remaining unconstrained generators will share the balance of the system load on an equal incremental cost basis. The final solution is obtained when successive solutions for each generator output PG_i vary less than the specified tolerance ESP2. The solution tolerances ESP1 and ESP2 are preset in the code to 0.001 but may be changed by the user.

Solution Method and Flow Chart

The steps used by the EDSP computer code to obtain the economic dispatch solution are as follows:

1. Input all data required for the load flow solution and add the normalized incremental cost curve coefficients for each generator and the individual generator power output limits.
2. Calculate the system transmission network admittance matrix [Y]. If a load flow analysis was previously performed, this admittance matrix will already be calculated and stored as "Y Bus Matrix." For the economic dispatch solution, the Y admittance matrix is inverted to obtain the "Z Bus Matrix." The Z Bus Matrix is also stored for subsequent runs and the real part of each element is used to determine r_{ik} in calculating the ITL_is.

3. Perform the load flow solution to obtain all bus parameters P_i, Q_i, V_i, and δ_i.
4. Calculate the individual incremental transmission losses ITL_i using the variables obtained from the load flow solution.
5. Select a starting value for lambda. The computer code arbitrarily sets the initial value of λ equal to the smallest incremental cost curve coefficient α.
6. Using the incremental cost curves, obtain the incremental cost and real power output of each generating unit corresponding to the selected value of λ.
7. Check that the real power output of each generator is within limits, i.e., $PG_{imin} \leq PG_i \leq PG_{imax}$.
8. Using Equation (3.25), check to see if the system real power balance is satisfied within the designated tolerance ESP1. If Equation (3.25) is not satisfied, the value of λ is adjusted and steps 6 and 7 repeated until a balance is obtained or the maximum number of iterations designated is reached.
9. Solution convergence has occurred when two successive power generation profiles vary less than the tolerance designated, i.e., ESP2.

$$PG_i^{p+1} - PG_i^{p} \leq ESP2 \quad i=1,2,\ldots n \tag{5.1}$$

If Equations (5.1) is not satisfied, the solution returns to step 3 for another load flow solution and rerun of steps 3 through 8 until the real power output at each generator varies less than ESP2 between iterations or the maximum number of iterations designated has been reached.

10. Calculate the individual transmission line power flow, the total system load and transmission losses, the individual generator operating costs and the total system operating cost using Equations (4.24) through (4.27), (3.20), and (3.24) respectively.

Figure 5.1 is a flow chart showing the principal steps in the economic dispatch solution method.

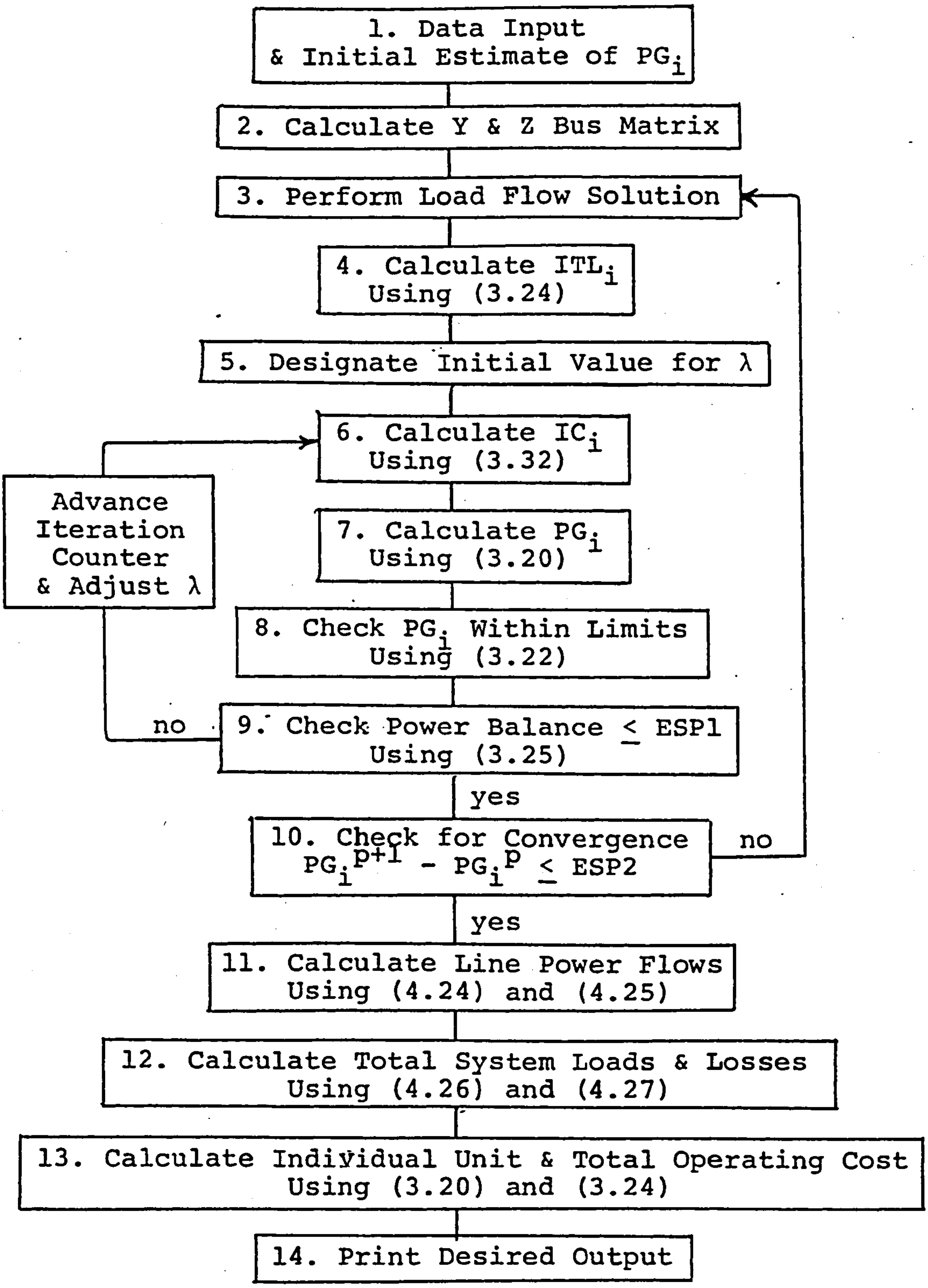

1. Data Input
& Initial Estimate of PG_i
2. Calculate Y & Z Bus Matrix
3. Perform Load Flow Solution
4. Calculate ITL_i
Using (3.24)
5. Designate Initial Value for λ
6. Calculate IC_i
Using (3.32)
Advance
Iteration
Counter
& Adjust λ
7. Calculate PG_i
Using (3.20)
8. Check PG_i Within Limits
Using (3.22)
no
9. Check Power Balance ≤ ESP1
Using (3.25)
yes
10. Check for Convergence
$PG_i^{p+1} - PG_i^p \leq$ ESP2
no
yes
11. Calculate Line Power Flows
Using (4.24) and (4.25)
12. Calculate Total System Loads & Losses
Using (4.26) and (4.27)
13. Calculate Individual Unit & Total Operating Cost
Using (3.20) and (3.24)
14. Print Desired Output

Figure 5.1 Economic Dispatch Solution Flow Chart

CHAPTER 6

TGE POWER SYSTEM

The parameters representing a power system transmission line network, bus or load center characteristics, and generator output cost curves have been derived. The computer code for load flow and economic dispatch solution methods has been developed. In this chapter the computer model will be applied to an actual power system, i.e., the TGE power system. The entire TGE system is described in terms of the parameters defined and computerized solutions of typical load flow and economic dispatch problems are obtained. The computer outputs of solution data for the sample problems are shown in Appendix A.

A single line schematic of the TGE system is shown in Figure 6.1. The network is divided into districts or load centers encircling the city of Tucson, Arizona and interconnected by 138KV high voltage transmission lines, i.e., North Loop, Northeast Loop, East Loop, Vail Loop, Ft. Huachuca Sub-station, South Loop, Tucson Loop, and the DeMoss Petrie and Irvington Generating Stations. The system is represented by 48 buses and 59 transmission lines with the control center and main generating plant located at the Irvington area. This generating station also

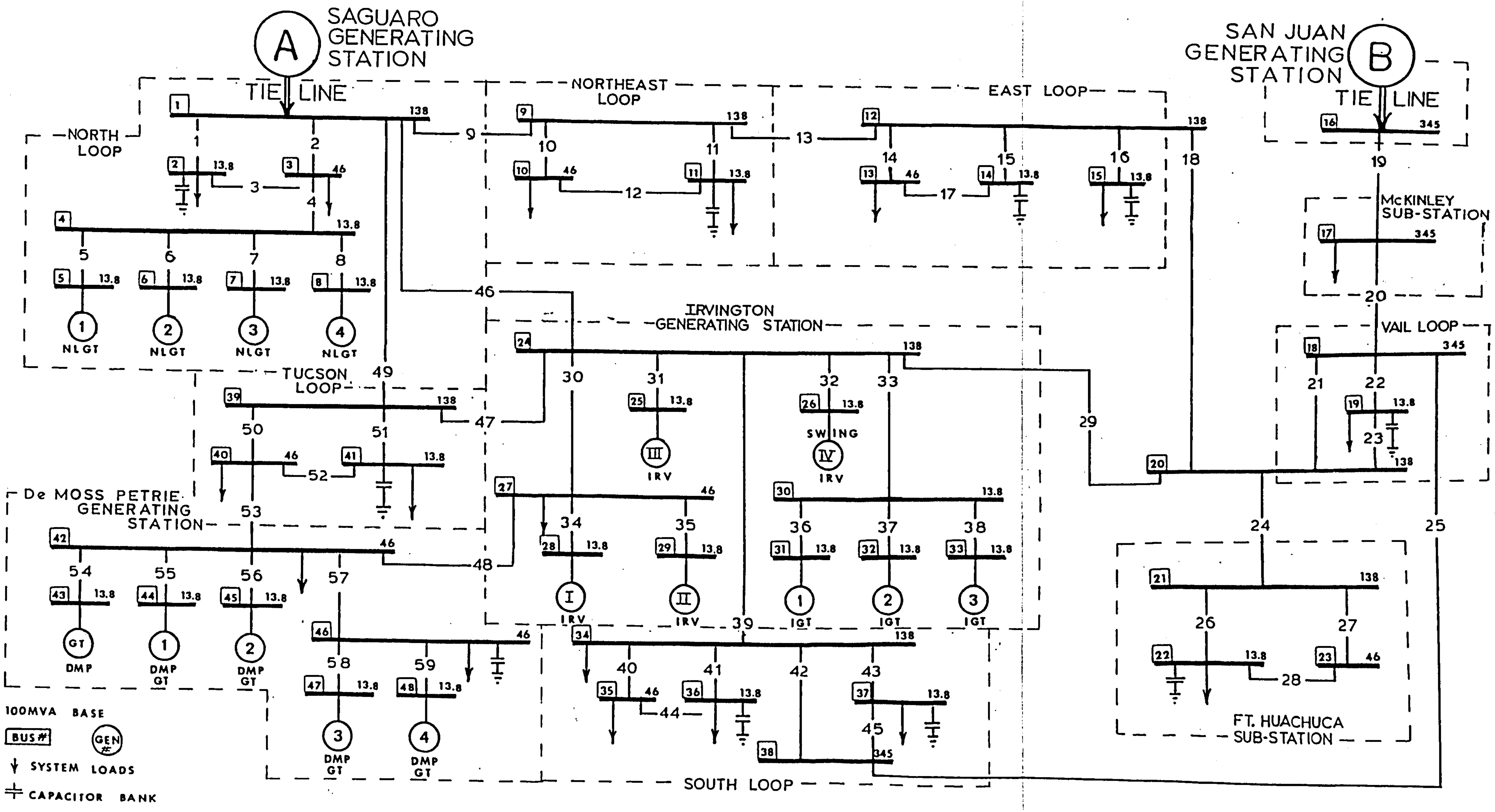


Figure 6.1 TGE Transmission System

contains the "slack" bus for the system. Two intra-area 345KV tie lines connect the TGE system with the Arizona Power System (APS) Saguaro generating station A and the joint owned San Juan generating station B as shown in Figure 6.1.

Each load center generally has three buses interconnected by a three winding transformer to provide three voltage levels as indicated on the right end of each bus, i.e., 138KV, 46KV, and 13.8KV. The transformers interconnecting these buses are usually a Y-Delta-Y configuration with the delta winding at 13.8KV. The transformers are not shown on the schematic but are represented as transmission lines with their equivalent network impedances shown as line data, e.g., lines 1, 2, and 3 at North Loop are transformer equivalent circuits. Bus numbers shown in the boxes at the left end of each bus and the transmission line numbers were assigned arbitrarily. Voltage compensating capacitor banks are shown as shunt capacitors to ground on the affected buses.

Since each area contains only one bus at each voltage level, the arrows attached to these buses represent a composite of all the loads in the area connected radially to the indicated bus. Lumping the individual loads and transmission losses together as one composite load does not give a true picture of total system transmission losses. However, power distribution at this level in the system is fixed and is not the primary concern of our economic dispatch study, i.e., there is no alternate routing to these loads to service them more economically. Planning studies where the economic advantages of adding additional alternate transmission lines to satisfy large power users would require the model to be expanded below the load center level in these areas.

Topping or peaking gas turbine generators are used in the North Loop, DeMoss Petrie, and Irvington areas to supply additional power during peak demand periods. They are shown as circles in the schematic. The generators shown as I, II, III, and IV are the main boiler fired generators at the Irvington plant. A 100MVA power base is used for the system with all parameters shown in per unit values.

TGE Line Data

Only lumped distribution network parameters were available from TGE; therefore, the length of all lines is shown as unity. However, the following cable types are typical of those used throughout the TGE system: ACSR #795MCM for 138KV lines, ACSR #477MCM or AA #4/0 for 46KV lines, and AA #1/0 or AA #4/0 for the 13.8 KV lines. The values used to represent the TGE transmission line distribution network are shown in Problem 3, Appendix A.

TGE Bus Data

A moderate total load for the system of 6.85 per units was assumed and distributed as shown in Problem 3 of Appendix A "Bus Data." The initial estimate for the generator power settings was arbitrarily selected and a flat voltage profile was used for each bus except 1 and 16. These buses receive tie line power and consequently were fixed at their maximum voltage levels. As explained in Chapter 7 tie line power at a fixed cost is used prior to local generation. All buses with compensating capacitor or inductor banks are designated type 2 buses to determine the amount of compensation required. Bus 26 at the Irvington Generating Station is the "slack" bus for the system. A negative bus type is used to indicate those buses with generators attached. The values used to represent the TGE bus data are shown in Problem 3, Appendix A.

Generator Data

Incremental heat rates for 9 typical generating units were obtained for the TGE system shown in Figure 6.1 in lieu of actual TGE performance data. The equation representing each heat curve and the location of each type turbine generator used is shown in Table 6.1. Y_i represents the ith turbine generator incremental heat rate in BTU per kilowatt hour and PG_i represents its net electrical output in megawatts. To convert these curves into incremental

Table 6.1 Typical Generator Incremental Heat Rates

Boiler Fired IRV I generator at Bus 28	
$Y_{28} = 8,809.36 + 10.01PG_{28} + 0.0441PG_{28}^{2}$	BTU/KWHr
Boiler Fired IRV II generator at Bus 29	
$Y_{29} = 7,948.05 + 27.8PG_{29} + 0.017PG_{29}^{2}$	BTU/KWHr
Boiler Fired IRV III generator at Bus 25	
$Y_{25} = 8,708.21 + 12.4PG_{25} + 0.0089PG_{25}^{2}$	BTU/KWHr
Boiler Fired IRV IV generator at Bus 26	
$Y_{26} = 8,399.7 + 4.205PG_{26} + 0.013PG_{26}^{2}$	BTU/KWHr
Gas Turbine Peaking generator at Buses indicated	
$Y_{43} = 9,461.7 + 45.22PG_{43} - 0.212PG_{43}^{2}$	BTU/KWHr
$Y_{44} = 9,209.8 + 138.7PG_{44} + 1.933PG_{44}^{2}$	BTU/KWHr
$Y_{45} = 9,209.8 + 128.7PG_{44} + 1.833PG_{45}^{2}$	BTU/KWHr
$Y_{48} = 9,511.74 + 52.22PG_{48} - 0.212PG_{48}^{2}$	BTU/KWHr
$Y = 9,561.74 + 55.22PG - 0.212PG^{2}$	BTU/KWHr
at Buses 5-8, 31-33, and 47	

cost curves, the fuel heating value (BTU per pound) and the cost (dollars per pound) of the fuel used is required. Current values of three types of fossile fuels are shown below so that a comparison of system operating cost can be made by type fuel (see, e.g., Friedlander, 1977; The Oil and Gas Journal, May 1, 1978:32; June 12, 1978:100-101)

Delivered Coal (e.g., Four Corners Coal)	112.0¢/10^6 BTU
Natural Gas (e.g., West South Central Gas)	147.4¢/10^6 BTU
Oil (e.g., West South Central Oil)	155.0¢/10^6 BTU

The normalized incremental cost curves are obtained by multiplying the incremental heat rate curve coefficients by the selected fuel cost in dollars per BTU and then using Equation (3.21) for the per unit value. A power base S_b of 100MVA was arbitrarily chosen for the system and three system cost bases C_b were used depending on the type fuel used. They are: $112.0/Hr for Coal, $147.4/Hr for natural gas, and $155.0/Hr for oil. The normalized incremental cost curve coefficients used for each generator and an arbitrarily selected set of generator constraints for each generator and type 2 bus are shown in Problem 3 of Appendix A.

A load flow and economic dispatch solution has been included in Problem 3 using the parameters shown as input data to describe the system shown in Figure 6.1. Actual performance data were not available for comparison at this time.

CHAPTER 7

COMPUTER CODE DESCRIPTION AND OPERATION

The computer code is a series of FORTRAN 10 statements for use with the DEC-10 FORTRAN 10 optimizing compiler. It has a main program and a series of subroutines to perform special repetitive functions. Two codes were developed: EDSP50 to accommodate a maximum of 50 buses and 100 transmission lines, and EDSP15 to accommodate 15 buses and 30 transmission lines. EDSP15 includes an accelerated Gauss-Seidell load flow solution method for instructional and demonstration purposes. The code can be changed easily to accommodate any size system by expanding the dimension statements. Both computer code listings are shown in Appendix B.

The computer code is designed as an interactive program and will prompt the user when action or input is required. The prompting messages are explained below.

))ENTER TYPE PROBLEM. A response of 0 or 1 is required for selection of a load flow or economic dispatch study respectively.

))ENTER LOAD FLOW SOLUTION METHOD. This message is received on the EDSP15 code only and permits load flow studies by Newton-Raphson or accelerated Gauss-Seidel methods. The EDSP50 code solves the load flow problem by Newton-Raphson methods only.

))ENTER THE TOTAL NUMBER OF (LINES, & BUSES).

))WANT TO TYPE IN NEW (LINE DATA, BUS DATA, GENERATOR DATA, OR OR CALCULATE Y/Z BUS MATRIX)?

A response of Y or N is required for each data group, i.e., "NYNN" would indicate you wish to enter new bus data only. Note that there are no spaces or commas between letters in the response. Also note that the letters must be capitalized. If the response is N for any data group, the necessary data must be stored in their respective temporary SOS files, i.e., for each N response the code looks in the respective SOS file as outlined in Chapter 2 for the necessary input data. When entering new data for the first time, the entries must be in the order prompted and each data entry must be separated by a space or a comma. Non-integer numbers such as any system parameter may be entered in floating point or exponential form but must be in per unit values. If the data are entered during program execution, temporary SOS data files as outlined in Chapter 2 are automatically created so that subsequent runs may be made using the same data without having to re-enter it for each run. For large programs it will probably be easier to enter data directly into the temporary SOS data files prior to execution of the code since editing or error corrections are easier and quicker with the SOS text editor.

If an entry is unacceptable, the code will so indicate and a corrected entry will be accepted. However, erroneous data will not be detected and correction of the error must wait until after the data group is completed.

When entering "Line Data," the transmission line parameters should be in per unit value per unit length.

When entering "Bus Data," the bus type must be negative to indicate the presence of a generator. Also, an initial estimate is required for each generator power PG_i and QG_i and the bus voltage V_i and δ_i not given. These initial estimated values are usually not critical but for very large problems, a poor initial estimate could increase the computation time or even cause divergence of the solution. The best guide is to use a flat voltage profile where $|V_i| \underline{/\delta_i}$ equals $1.0\underline{/0.0}$ for the unknown voltages and $\frac{\Sigma PD_i}{\text{No. of Gen.}}$ and $\frac{\Sigma QD_i}{\text{No. of Gen.}}$ equals the estimated real and reactive powers respectively. The EDSP15 code automatically sets δ_i to 0.0 initially while the user must specify $|V_i|$.

When entering "Generator Data," provisions have been included for accepting "Tie Line" power where the generator cost curves are not applicable. In this case, the generator cost curve coefficients should be entered as 0.0 for each coefficient alpha, beta, and gamma indicating constant cost tie line power. The code schedules the use of this power prior to use of locally generated power.

When entering "Generator Constraints," the code requires the following:

1. For load flow problems only: reactive power limits are required for Type ± 2, voltage control buses, i.e., QG_{imax} and QG_{imin}. Real power limits PG_{imax} and PG_{imin} are not used.
2. For economic dispatch problems: the real power limits PG_{imax} and PG_{imin} are required for all generators including "Tie Line" power. The reactive power limits QG_{imax} and QG_{imin} are also required for each Type ± 2, voltage control bus.

The real and reactive power limits can be used to fix the generated power at a specific level by setting the maximum and minimum limits at the same value.

No entries are required for convergence criteria. They are pre-set in the computer code as follows:

1. The maximum number of load flow iteration permitted for the Newton-Raphson solution method is 5. In the EDSP15 code the maximum number of iterations permitted for the Gauss-Seidel solution method is 30.
2. The load flow solution tolerance TOL is set at 0.0001.
3. The maximum number of economic dispatch iterations ITER for determination of λ and to obtain a power balance in the system is set at 50.

4. The economic dispatch tolerance ESP1 for the system power balance, Equation (3.25), is set at 0.001.

5. The economic dispatch solution tolerance ESP2 for convergence is set at 0.001, i.e., Equation (5.1).

If values other than these "pre-set" values are desired, they may be revised when prompted during data change requests.

))WANT A COPY OF (LDATA, BDATA, GDATA, OR SYSTEM CONVERGENCE CRITERIA?) A response of N or Y is required for each data group for which a printout is desired, i.e., YNNN will cause printout of "Line Data" only.

))WANT TO CHANGE (LDATA, BDATA, GDATA, OR CONVERGENCE CRITERIA)? Again a response of N or Y is required for each data group where data changes are desired, i.e., YNNY would permit changes to both "Line Data" and "Convergence Criteria." In this mode, entries may be corrected as many times as desired. Further, the changes may be posted to the temporary SOS files if desired. The EDSP15 code automatically updates the SOS file with the changes when the data group is exited by typing 0.

It should be realized that changes in one data group may require changes in another. For instance, a change of a bus type from 1 to 2 in "Bus Data" would require the addition of reactive power limits at this bus in the

"Generator Data" group. Another example is shown in Problem 2, Appendix A, when changing from a load flow study to an economic dispatch analysis, the generator cost curve coefficients must be added and the Z bus matrix calculated. Also, if the total number of buses or lines is increased, the additional line, bus, and generator parameters may be added by requesting a change in their respective data groups.

```
))WANT A COPY OF (Y BUS MATRIX, OR Z BUS MATRIX)?
```

A Y or N is required for each data group. If a load flow problem is being performed the Z Bus matrix is not calculated and therefore will not print out.

```
))TYPE CODE # FOR DESIRED ACTION
    1=PRINT BUS & GEN SOLUTIONS ONLY
    2=PRINT LINE POWERS, BUS & GEN SOLUTIONS
    3=CREAT SOS FILE OF SOLUTIONS
    4=CHANGE DATA OR TOLERANCES AND RERUN
    5=CHANGE TYPE PROBLEM OR SOLUTION METHOD,
        RERUN WITH EXISTING DATA
    6=NEW PROBLEM
    7=TERMINATE
```

An entry of 1 or 2 will obtain a printout of the solution data indicated at the user's terminal. An entry of 3 will create a temporary SOS data file of bus, generator, and line power flow solutions so that all or part of the data may be printed on the high speed line printer. Also, the temporary data and solution files may be transferred to magnetic tape for permanent storage. Entering 4 or 5 will permit changes of type problem or input data using the previous solution as an initial starting point for the new calculations. For large programs this permits considerable savings in computer calculation time when only small changes to the input data are desired. Since this is less significant for small size systems, EDSP15 re-enters the original input data from the SOS files prior to each rerun. Items 6 and 7 are self explanatory.

Diagnostics

During load flow calculations the following messages may be transmitted to inform the user when programmed limits have been reached:

"Bus #___ exceeds QGMAX/QGMIN, bus changed to type 1." This message applies to Type 2 control buses only. As indicated, it informs the user that the specified reactive power limits have been reached. The type bus is then changed to a Type 1 and the solution continues letting the voltage magnitude vary until a solution is obtained or the maximum number of iterations has been reached. The buses that were changed to Type 1 are returned to a Type 2 and the bus voltage magnitudes are reset to 1.0 if a rerun of the problem is requested, i.e., action 4 or 5. Also, the bus type and voltage magnitude are reset to type 2 and 1.0 respectively between economic dispatch iterations. This permits "back-off" from limits at these buses during solution for the optimum power settings.

"No convergence in ___ iterations." This message indicates the specified solution tolerance could not be obtained within the number of iterations permitted. The calculations to this point in the solution should be printed for study before a rerun is accomplished with an increased number of iterations.

During economic dispatch calculations the following messages may be received:

"System power loss is negative, check total system power generation." This is a check to see that the power balance equation (3.25) was satisfied during load flow calculations.

"Maximum constant cost power not used, load flow problem only." This indicates that the total system load plus losses is less than the amount of power available from the tie line power. As stated earlier, constant cost tie line power is used prior to locally generated power.

"Power balance equation not satisfied in ___ iterations." This indicates that the maximum number of economic dispatch iterations was reached before Equation (3.25) could be satisfied by varying Lambda. The real power generation limits, cost curve coefficients, and convergence criteria should be reviewed before a rerun is accomplished. Again a printout of the bus and generator solutions will reveal calculation results to that point in the solution process and will probably indicate the problem area.

CHAPTER 8

RESULTS AND CONCLUSIONS

Results

This chapter will include the findings and significant observations made during the use of the computer code. While space considerations preclude the inclusion of machine runs to demonstrate each of our observations, sample problems were provided in Appendix A for familiarization with the computer code mechanics. Hopefully, the users will then develop their own power system analysis techniques with the observations included.

As stated in Chapter 4, the Newton-Raphson solution method was chosen for the large computer code EDSP50. Initial evaluation of load flow solution methods showed that the Gauss-Seidel or successive displacement method had many conditions for which a solution could not be obtained, i.e., the solution would diverge, oscillate, or converge to a point and go no further. Also, the number of iterations increased rapidly as the problem size increased. Therefore, the Gauss-Seidel solution method was retained in the smaller code EDSP15 only. It contains an acceleration factor of 1.5 in the solution algorithm to increase the speed of convergence. That is, the change in voltage magnitude and phase angle between each iteration is multiplied by the acceleration factor of 1.5. Since speed of convergence is influenced by the size of the system, impedance characteristics of the transmission network, and how heavily the system is loaded, there may not be an optimum acceleration factor for this solution method. Further study may develop a nonlinear acceleration factor to improve this method but since the Newton-Raphson methods proved superior in convergence reliability, no further effort was expended on the Gauss-Seidel method.

The computer code description in Chapter 7 along with the sample problems should familiarize the user with the operation of the code and its options for the control of power system studies. To enhance the utility of the code to the power system engineer, a few of the more significant results of power system studies are included below. They become important as the size of the system increases, e.g., above 25 buses.

1. The "slack" bus should be near the electrical center of the system.
2. In larger systems, as the load increases, voltage control at certain buses becomes critical. Therefore, numerous type 2 buses should be used throughout the system. This in effect places additional reactive power "slack" buses in the system.

3. Depending on the load conditions, electrical characteristics of the transmission network, and the generator cost curve characteristics, the lowest total system operating cost may not be the lowest total transmission loss operating point, e.g., problems 1 and 2 in Appendix A.

4. If convergence fails, the bus, generator, and line power flow calculations at the point of solution termination should be printed out for study before reruns with lower solution tolerances or increase iteration limits are accomplished. The area of difficulty can usually be detected. For example, if the phase angle at a particular bus exceeds 1.5, the maximum transmission line capacity is being exceeded and system convergence will fail due to the inability to satisfy the power balance equation (3.16). Also, if bus voltage is too low or too high the reactive power generation and limits should be checked. If during economic dispatch analysis, the load flow calculations converge but the economic dispatch solution does not converge, the generator cost curves and real power generation limits should be checked. Solution tolerance and iteration limits can then be relaxed if necessary to obtain solution convergence. However, if the load flow solution is not accomplished in 3 to 5 iterations there is

usually an error in data entry or the initial estimate of system parameters. These should be rechecked before reruns with expanded iteration limits are made.

Conclusions

The computer code developed is an effective engineering tool for the analysis of power system performance. Its direct on-line calculations and interactive capability provide the flexibility to significantly reduce the engineering manhours involved in problem set-up and system analysis. In addition, its ability to store network parameters for use in subsequent studies further enhances its utility for quick response problems such as serious outage conditions. The code has demonstrated sufficient accuracy to reliably determine system transmission losses and operating cost data. Also, this code can be used on nuclear or fossile fuel powered generating plants but would not be applicable to hydroelectric plants unless a fuel equivalent ratio were developed for the water used in electric power generation. The timesharing feature of the DEC-10 provides a significant cost advantage in that a high cost computer need not be dedicated to the sole support of the engineering effort but can serve all management functions simultaneously.

Current nationwide energy conservation efforts dictate that electrical power companies continue their efforts to improve efficiency. The latest industry survey by Friedlander (1977:51) indicates that 50 per cent of the power companies have now developed some type of system performance analysis capability. The computer code developed in this thesis continues this effort to optimize the analysis process for continued improvement in energy conservation efforts.

Recommendations

Recent advances in computer technology have expanded the capability of digital computers to process and store the large quantity of data necessary to analyze electrical power system performance. Therefore, the power system engineer should continue to develop new and better ways to use this increasing computer capacity for the improvement of all phases of electrical power production and distribution. Research should continue in the area of interactive computer programming of electrical power system performance and control. Extensive use of this interactive code indicates the following areas where further development could enhance its utility and capability.

1. Replace the matrix inversion routines used to obtain Z bus matrix and the inverse jacobian matrix in Equations (4.16) and (4.17) with a direct

solution method using optimally ordered triangular factorization (see, e.g., Tinney and Walker, 1967: 1801-1809; Tinney and Meyer, 1973). This would reduce the computer time required to obtain these inverse matrices by one-third. It would also improve solution accuracy by reducing roundoff errors inherent in matrix inversion routines. In addition, it would preserve the sparcity that is characteristic of the Y admittance matrix. This would permit further reduction in central (core) memory storage requirements (see, e.g., Ogbuobiri, 1970:150-155; Tinney and Sato, 1963).

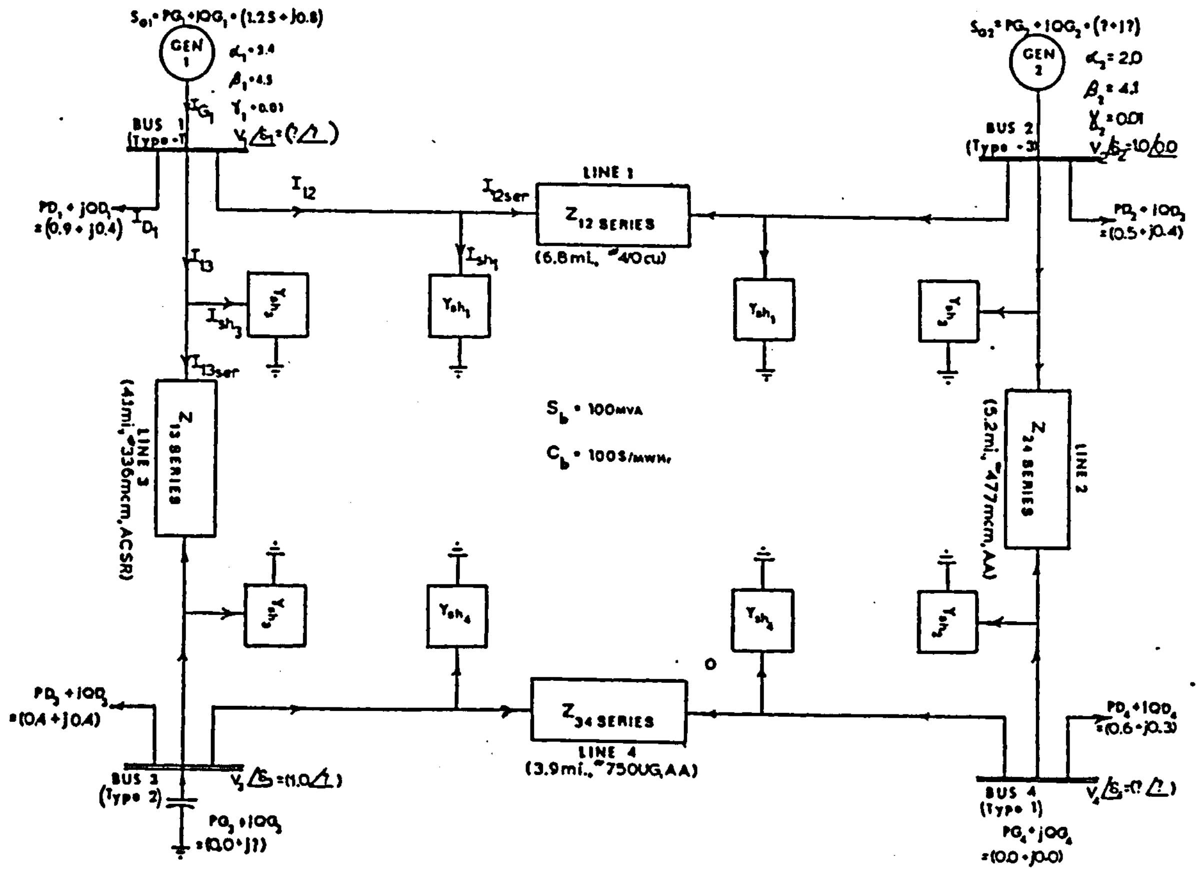
2. Features could be added to accommodate tap changing under load and phase shifting transformers (see, e.g., Elgerd, 1971:263-270; Peterson and Meyer, 1971:103-108). This would improve flexibility in voltage control and line power flow and further enhance the solution convergence reliability.

3. The incorporation of penalty factors instead of hard limits for reactive power generation and steepest descent or gradient method for finding the optimum power point should be investigated as ways to further improve speed and reliability of convergence (see, e.g., Elgerd, 1971:304-312; Dommel and Tinney, 1968; Peschon et al., 1968:40-48).

APPENDIX A

SAMPLE PROBLEMS

Problem 1: 4-Bus Power System



```
.RUN EDSP15[3733,547]

               ****ECONOMIC DISPATCH EXECUTIVE PROGRAM****

>>ENTER TYPE PROBLEM
       0=LOAD FLOW PROBLEM
       1=ECONOMIC DISPATCH PROBLEM
0

>>ENTER LOAD FLOW SOLUTION METHOD
       0=NEWTON-RAPHSON
       1=ACCELERATED GAUSS-SEIDEL
       2=PATTERN SEARCH
0

>>ENTER TOTAL NUMBER OF (LINES, & BUSES).
4,4

>>WANT TO TYPE IN NEW(LINE DATA, BUS DATA, OR GENERATOR DATA)?
YNY

**SYSTEM LINE DATA**
>>ENTER (LINE #,SB #,EB #, LINE LENGTH, YSHUNT G,YSHUNT B,ZSERIES R,ZSERIES X)
1,1,2,35.904,0.0,0.0,0.271267E-2,0.60586E-2
2,2,4,27.456,0.0,0.0,0.19329E-2,0.56475E-2
3,1,3,21.648,0.0,0.0,0.27410E-2,0.56522E-2
4,3,4,20.592,0.0,9.2531E-4,0.15123E-2,0.17958E-2

**SYSTEM BUS DATA**
>>ENTER(BUS#, BUS TYPE,PG,QG,PD,QD,VSPEC)
1,-1,1.25,0.8,0.9,0.4,1.0,0.0\0.0,\
2,-3,1.2,1.1,0.5,0.4,1.0
3,2,0.0,0.4,0.4,0.4,1.0
4,1,0.0,0.0,0.6,0.3,1.0

**GENERATOR CONSTRAINT DATA**
>>ENTER FOR TYPE  -1 & -3  BUS (BUS #,PGMAX,PGMIN)
        FOR TYPE  -2 BUS       (BUS #,PGMAX,PGMIN,QGMAX,QGMIN)
        FOR TYPE  +2 BUS       (BUS #,QGMAX,QGMIN)
1,3.0,0.5
2,3.0,0.5,2.0,-2.0
3,2.0,-2.0

>>WANT A COPY OF (LDATA,BDATA,GDATA,OR SYSTEM CONVERGENCE CRITERIA)?
YYYY

**SYSTEM LINE DATA**
LINE # SB EB    LENGTH               Y SHUNT                          Z SERIES
  1     1  2  0.35904E+02  0.00000E+00 0.00000E+00  0.27127E-02 0.60586E-02
  2     2  4  0.27456E+02  0.00000E+00 0.00000E+00  0.19329E-02 0.56475E-02
  3     1  3  0.21648E+02  0.00000E+00 0.00000E+00  0.27410E-02 0.56522E-02
  4     3  4  0.20592E+02  0.00000E+00 0.92531E-03  0.15123E-02 0.17958E-02

**SYSTEM BUS DATA**
 BUS   TYPE      PG          QG          PD          QD        VSPEC
  1     -1    1.25000    0.80000     0.90000     0.40000     1.00000
  2     -3    1.20000    1.10000     0.50000     0.40000     1.00000
  3      2    0.00000    0.40000     0.40000     0.40000     1.00000
  4      1    0.00000    0.00000     0.60000     0.30000     1.00000
```

```
**GENERATOR CONSTRAINTS**
BUS #    PGMAX      PGMIN        QGMAX       QGMIN
  1      3.0000      0.5000
  2      3.0000      0.5000
  3                              2.0000     -2.0000

**SYSTEM CONVERGENCE CRITERIA**
         LOAD FLOW ITERATIONS=  5
         LOAD FLOW TOLERANCE=  0.10000E-03
         ECON DISP ITERATIONS= 50
         ESP1=  0.10000E-02
         ESP2=  0.10000E-02

>>WANT TO CHANGE (LDATA,BDATA,GDATA, OR CONVERGENCE CRITERIA)?
YYYY

** LINE DATA CHANGES**
>>ENTER (VARIABLE #, LINE #, NEW VALUE THEN 'CR')
         1=STARTING BUS #
         2=ENDING BUS #
         3=LINE LENGTH
         4=Y SHUNT G
         5=Y SHUNT B
         6=Z SERIES R
         7=Z SERIES X
         8=RETYPE COPY OF ALL LINE DATA
         9=CHANGE TOTAL NUMBER OF LINES
         >>ENTER 0 THEN'CR' WHEN CHANGES COMPLETED
0

**BUS DATA CHANGES**
>>ENTER(VARIABLE #, BUS #, NEW VALUE THEN 'CR'
         1=BUS TYPE
         2=PG
         3=QG
         4=PD
         5=QD
         6=VSPEC
         7=RETYPE COPY OF ALL BUS DATA
         8=CHANGE TOTAL NUMBER OF BUSES
         >>ENTER 0 THEN 'CR' WHEN CHANGES COMPLETED.
0

**GENERATOR DATA CHANGES**
>>ENTER (VARIABLE #, BUS #, NEW VALUE THEN 'CR')
         1=ALPHA
         2=BETA
         3=GAMMA
         4=PGMAX
         5=PGMIN
         6=QGMAX
         7=QGMIN
         8=RETYPE ALL GENERATOR DATA
         >>ENTER 0 THEN 'CR' WHEN CHANGES COMPLETED.
0
```

```
**CONVERGENCE CRITERIA CHANGES**
>>ENTER(VARIABLE #,NEW VALUE THEN 'CR')
         1=LOAD FLOW ITERATIONS
         2=LOAD FLOW TOLERANCE
         3=ECON DISP ITERATIONS
         4=ESP1
         5=ESP2
         >>ENTER 0 THEN 'CR' WHEN CHANGES COMPLETED.
0

**GENERATING YBUS MATRIX**

>>WANT A COPY OF (YBUS MATRIX, ZBUS MATRIX)?
YN

**CALCULATED SYSTEM LINE DATA**
LINE #                Y SHUNT                         Y SERIES
  1      0.000000E+00  0.000000E+00     0.171459E+01 -0.382941E+01
  2      0.000000E+00  0.000000E+00     0.197584E+01 -0.577296E+01
  3      0.000000E+00  0.000000E+00     0.320870E+01 -0.661664E+01
  4      0.000000E+00  0.952699E-02     0.133240E+02 -0.158217E+02

**YBUS MATRIX**
ROW  1
0.11548E+02-0.11304E+01 0.41957E+01 0.19918E+01 0.73536E+01 0.20223E+01
0.00000E+00 0.00000E+00
ROW  2
0.41957E+01 0.19918E+01 0.10287E+02-0.12039E+01 0.00000E+00 0.00000E+00
0.61017E+01 0.19006E+01
ROW  3
0.73536E+01 0.20223E+01 0.00000E+00 0.00000E+00 0.27864E+02-0.93559E+00
0.20685E+02 0.22707E+01
ROW  4
0.00000E+00 0.00000E+00 0.61017E+01 0.19006E+01 0.20685E+02 0.22707E+01
0.26458E+02-0.95418E+00

**SOLVING LOAD FLOW EQUATIONS BY  NR**
****NR CONVERGENCE IN   3 ITERATIONS

>>TYPE CODE # FOR DESIRED ACTION
       1=PRINT BUS & GEN SOLUTIONS ONLY.
       2=PRINT LINE  POWERS, BUS & GEN SOLUTIONS
       3=CREATE SOS FILE OF  SOLUTIONS
       4=CHANGE DATA OR TOLERANCES AND RERUN
       5=CHANGE TYPE PROBLEM OR SOLUTION METHOD,
            RERUN WITH EXISTING DATA
       6=NEW PROBLEM
       7=TERMINATE
2

****SYSTEM BUS AND GENERATOR SOLUTIONS****
SYSTEM LOSSES(PU)=  0.0431,  SYSTEM LOAD(PU)=  2.4000
BUS # BTYPE       PG      QG      PD      QD     VMAG    VANG
  1    -1      1.2500  0.8000  0.9000  0.4000  1.0419 -0.0521
  2    -3      1.1931  0.0965  0.5000  0.4000  1.0000  0.0000
  3     2      0.0000  0.6846  0.4000  0.4000  1.0000 -0.1008
  4     1      0.0000  0.0000  0.6000  0.3000  0.9832 -0.0906
```

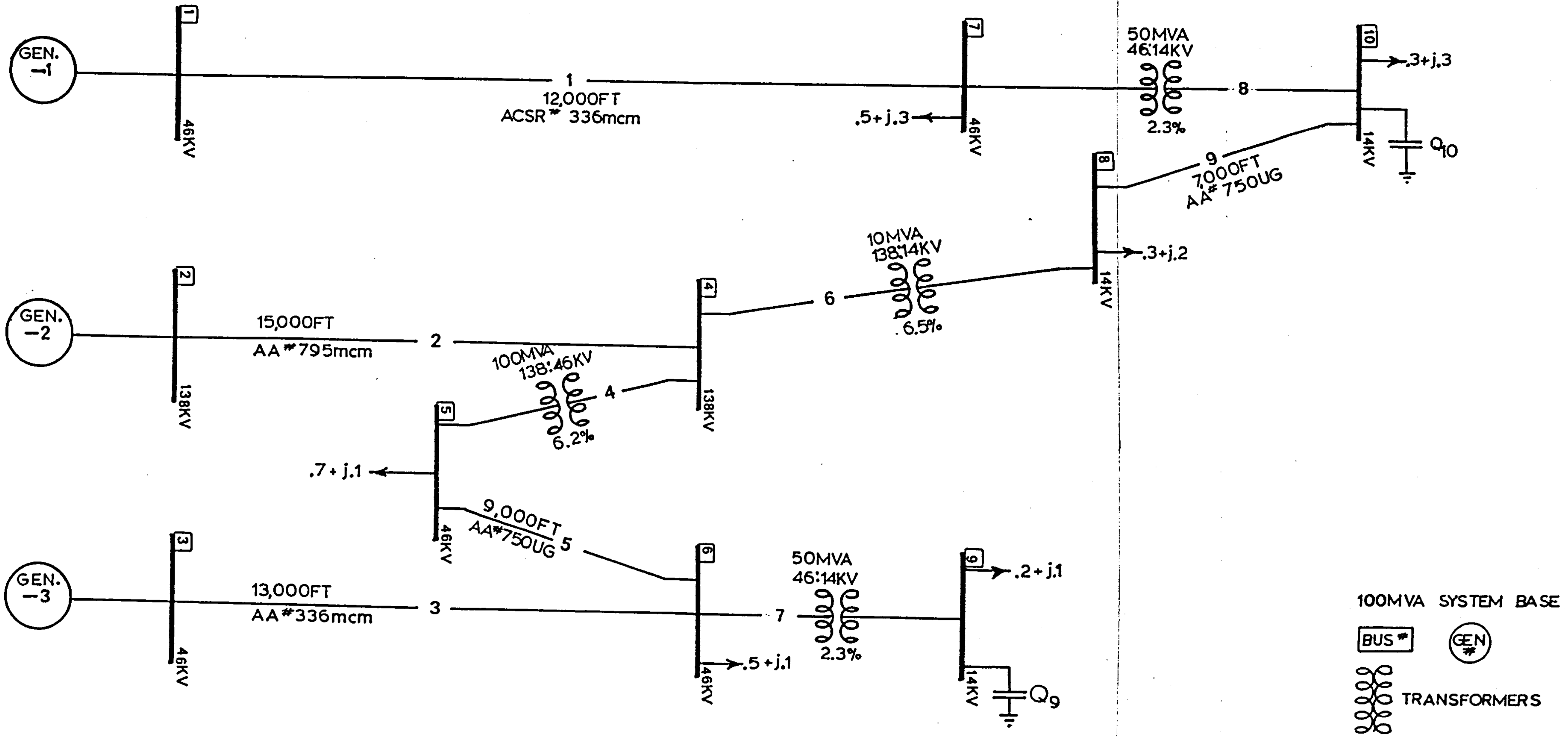
```
****LINE POWER FLOW****
LINE#    SB     EB     REAL POWER     REACTIVE POWER
  1       1      2      -0.1303            0.2657
  1       2      1       0.1381           -0.2481
  2       2      4       0.5550           -0.0554
  2       4      2      -0.5385            0.1036
  3       1      3       0.4803            0.1343
  3       3      1      -0.4667           -0.1063
  4       3      4       0.0667            0.3909
  4       4      3      -0.0615           -0.4036

>>TYPE CODE # FOR DESIRED ACTION
       1=PRINT BUS & GEN SOLUTIONS ONLY.
       2=PRINT LINE  POWERS, BUS & GEN SOLUTIONS
       3=CREATE SOS FILE OF  SOLUTIONS
       4=CHANGE DATA OR TOLERANCES AND RERUN
       5=CHANGE TYPE PROBLEM OR SOLUTION METHOD,
            RERUN WITH EXISTING DATA
       6=NEW PROBLEM
       7=TERMINATE
5
              ****ECONOMIC DISPATCH EXECUTIVE PROGRAM****

>>ENTER TYPE PROBLEM
       0=LOAD FLOW PROBLEM
       1=ECONOMIC DISPATCH PROBLEM
1

>>IS SYSTEM LOSSLESS?
NO

>>ENTER LOAD FLOW SOLUTION METHOD
       0=NEWTON-RAPHSON
       1=ACCELERATED GAUSS-SEIDEL
       2=PATTERN SEARCH
0

>>ENTER TOTAL NUMBER OF (LINES, & BUSES).
4,4

>>WANT TO TYPE IN NEW(LINE DATA, BUS DATA, OR GENERATOR DATA)?
NNN

>>WANT A COPY OF (LDATA,BDATA,GDATA,OR SYSTEM CONVERGENCE CRITERIA)?
NNYN

**SYSTEM GENERATOR DATA**
NUMBER OF GENERATORS =  2
BUS#     ALPHA      BETA       GAMMA
  1    2.40000   4.50000   0.01000
  2    2.00000   4.10000   0.01000

**GENERATOR CONSTRAINTS**
BUS #    PGMAX       PGMIN        QGMAX       QGMIN
  1      3.0000      0.5000
  2      3.0000      0.5000
  3                               2.0000     -2.0000

>>WANT TO CHANGE (LDATA,BDATA,GDATA, OR CONVERGENCE CRITERIA)?
NNNN
```

```
**GENERATING YBUS MATRIX**
              **CALCULATING ZBUS MATRIX**

>>WANT A COPY OF (YBUS MATRIX, ZBUS MATRIX)?
NY

**ZBUS MATRIX**
ROW  1
0.52388E+02-0.15700E+01 0.52446E+02-0.15705E+01 0.52478E+02-0.15707E+01
0.52487E+02-0.15709E+01
ROW  2
0.52446E+02-0.15705E+01 0.52375E+02-0.15700E+01 0.52486E+02-0.15709E+01
0.52479E+02-0.15707E+01
ROW  3
0.52478E+02-0.15707E+01 0.52486E+02-0.15709E+01 0.52474E+02-0.15707E+01
0.52491E+02-0.15709E+01
ROW  4
0.52487E+02-0.15709E+01 0.52479E+02-0.15707E+01 0.52491E+02-0.15709E+01
0.52474E+02-0.15707E+01

**SOLVING LOAD FLOW EQUATIONS BY  NR**
****NR CONVERGENCE IN   3 ITERATIONS
    ***OPTIMIZING COST FUNCTIONS***
          LAMBDA DETERMINED IN  15  ITERATIONS

**SOLVING LOAD FLOW EQUATIONS BY  NR**
****NR CONVERGENCE IN   2 ITERATIONS
    ***OPTIMIZING COST FUNCTIONS***
          LAMBDA DETERMINED IN   4  ITERATIONS

**SOLVING LOAD FLOW EQUATIONS BY  NR**
****NR CONVERGENCE IN  1 ITERATIONS
    ***OPTIMIZING COST FUNCTIONS***
          LAMBDA DETERMINED IN   0  ITERATIONS

**SOLVING LOAD FLOW EQUATIONS BY  NR**
****NR CONVERGENCE IN  1 ITERATIONS
    ***OPTIMIZING COST FUNCTIONS***
          LAMBDA DETERMINED IN   0  ITERATIONS

>>TYPE CODE # FOR DESIRED ACTION
       1=PRINT BUS & GEN SOLUTIONS ONLY.
       2=PRINT LINE  POWERS, BUS & GEN SOLUTIONS,
           AND SYSTEMOPERATING COST.
       3=CREATE SOS FILE OF  SOLUTIONS
       4=CHANGE DATA OR TOLERANCES AND RERUN
       5=CHANGE TYPE PROBLEM OR SOLUTION METHOD,
           RERUN WITH EXISTING DATA
       6=NEW PROBLEM
       7=TERMINATE
2

****SYSTEM BUS AND GENERATOR SOLUTIONS****
     ECON. DISPATCH CONVERGENCE IN  4 ITERATIONS
 ESP1=  0.0010, ESP2=  0.0010,  LAMBDA=  7.9094
SYSTEM LOSSES(PU)=  0.0464,  SYSTEM LOAD(PU)=  2.4000
BUS# BTYPE     IC      ITL      PG       QG       PD       QD      VMAG     VANG
  1  -1      7.5834  0.0412  1.1488  0.8000  0.9000  0.4000  1.0384 -0.0658
  2  -3      7.3368  0.0724  1.2976  0.0760  0.5000  0.4000  1.0000  0.0000
  3   2      0.0000  0.0000  0.0000  0.7144  0.4000  0.4000  1.0000 -0.1105
  4   1      0.0000  0.0000  0.0000  0.0000  0.6000  0.3000  0.9836 -0.0981
```

```
****SYSTEM OPERATING COST****
GEN.#  1=     5.73P.U.
GEN.#  2=     6.05P.U.
TOTAL SYSTEM OPERATING COST =      11.79P.U.

****LINE POWER FLOW****
LINE#    SB     EB     REAL POWER     REACTIVE POWER
  1       1      2      -0.1893             0.2783
  1       2      1       0.1995            -0.2554
  2       2      4       0.5981            -0.0686
  2       4      2      -0.5788             0.1248
  3       1      3       0.4381             0.1217
  3       3      1      -0.4267            -0.0982
  4       3      4       0.0267             0.4127
  4       4      3      -0.0212            -0.4248

>>TYPE CODE # FOR DESIRED ACTION
        1=PRINT BUS & GEN SOLUTIONS ONLY.
        2=PRINT LINE  POWERS, BUS & GEN SOLUTIONS,
             AND SYSTEMOPERATING COST.
        3=CREATE SOS FILE OF  SOLUTIONS
        4=CHANGE DATA OR TOLERANCES AND RERUN
        5=CHANGE TYPE PROBLEM OR SOLUTION METHOD,
             RERUN WITH EXISTING DATA
        6=NEW PROBLEM
        7=TERMINATE
4

>>WANT TO CHANGE (LDATA,BDATA,GDATA, OR CONVERGENCE CRITERIA)?
NNYN

**GENERATOR DATA CHANGES**
>>ENTER (VARIABLE #, BUS #, NEW VALUE THEN 'CR')
          1=ALPHA
          2=BETA
          3=GAMMA
          4=PGMAX
          5=PGMIN
          6=QGMAX
          7=QGMIN
          8=RETYPE ALL GENERATOR DATA
          9=UPDATE SOS FILE WITH CHANGES
          >>ENTER 0 THEN 'CR' WHEN CHANGES COMPLETED.
4,1,1.25
PGMAX(BUS #  1)= 0.12500E+01
5,1,1.25
PGMIN(BUS #  1)= 0.12500E+01


>>WANT A COPY OF (YBUS MATRIX, ZBUS MATRIX)?
NN

**SOLVING LOAD FLOW EQUATIONS BY  NR**
****NR CONVERGENCE IN  0 ITERATIONS
    ***OPTIMIZING COST FUNCTIONS***
          LAMBDA DETERMINED IN  13  ITERATIONS

**SOLVING LOAD FLOW EQUATIONS BY  NR**
****NR CONVERGENCE IN  2 ITERATIONS
    ***OPTIMIZING COST FUNCTIONS***
          LAMBDA DETERMINED IN   5  ITERATIONS
```

```
**SOLVING LOAD FLOW EQUATIONS BY  NR**
****NR CONVERGENCE IN  1 ITERATIONS
    ***OPTIMIZING COST FUNCTIONS***
        . LAMBDA DETERMINED IN    0  ITERATIONS

>>TYPE CODE # FOR DESIRED ACTION
       1=PRINT BUS & GEN SOLUTIONS ONLY.
       2=PRINT LINE  POWERS, BUS & GEN SOLUTIONS,
            AND SYSTEMOPERATING COST.
       3=CREATE SOS FILE OF  SOLUTIONS
       4=CHANGE DATA OR TOLERANCES AND RERUN
       5=CHANGE TYPE PROBLEM OR SOLUTION METHOD,
            RERUN WITH EXISTING DATA
       6=NEW PROBLEM
       7=TERMINATE
2

****SYSTEM BUS AND GENERATOR SOLUTIONS****
     ECON. DISPATCH CONVERGENCE IN  3 ITERATIONS
 ESP1=  0.0010, ESP2=  0.0010,  LAMBDA=  7.4000
SYSTEM LOSSES(PU)=  0.0431,  SYSTEM LOAD(PU)=  2.4000
BUS# BTYPE     IC      ITL      PG      QG      PD      QD      VMAG    VANG
  1  -1    7.0549  0.0466  1.2500  0.8000  0.9000  0.4000  1.0419 -0.0521
  2  -3    6.9078  0.0665  1.1931  0.0965  0.5000  0.4000  1.0000  0.0000
  3   2    0.0000  0.0000  0.0000  0.6846  0.4000  0.4000  1.0000 -0.1008
  4   1    0.0000  0.0000  0.0000  0.0000  0.6000  0.3000  0.9832 -0.0906

****SYSTEM OPERATING COST****
GEN.#  1=     6.52P.U.
GEN.#  2=     5.31P.U.
TOTAL SYSTEM OPERATING COST =      11.83P.U.

****LINE POWER FLOW****
LINE#    SB     EB     REAL POWER     REACTIVE POWER
  1      1      2       -0.1303             0.2657
  1      2      1        0.1381            -0.2481
  2      2      4        0.5550            -0.0554
  2      4      2       -0.5385             0.1036
  3      1      3        0.4803             0.1343
  3      3      1       -0.4667            -0.1063
  4      3      4        0.0667             0.3909
  4      4      3       -0.0615            -0.4036

>>TYPE CODE # FOR DESIRED ACTION
       1=PRINT BUS & GEN SOLUTIONS ONLY.
       2=PRINT LINE  POWERS, BUS & GEN SOLUTIONS,
            AND SYSTEMOPERATING COST.
       3=CREATE SOS FILE OF  SOLUTIONS
       4=CHANGE DATA OR TOLERANCES AND RERUN
       5=CHANGE TYPE PROBLEM OR SOLUTION METHOD,
            RERUN WITH EXISTING DATA
       6=NEW PROBLEM
       7=TERMINATE
7
STOP

END OF EXECUTION
CPU TIME: 1.54  ELAPSED TIME: 5:46.85
EXIT
```

Problem 2: 10-Bus Power System



```
RUN EDSP25[3733,547]

              ****ECONOMIC DISPATCH EXECUTIVE PROGRAM****

>>ENTER TYPE PROBLEM
       0=LOAD FLOW PROBLEM
       1=ECONOMIC DISPATCH PROBLEM
0

>>ENTER TOTAL NUMBER OF (LINES, & BUSES).
9,10

>>WANT TO TYPE IN NEW(LINE DATA, BUS DATA, OR GENERATOR DATA,
    OR CALCULATE Y/ZBUS MATRIX)?
NNNY

>>WANT A COPY OF (LDATA,BDATA,GDATA,OR SYSTEM CONVERGENCE CRITERIA)?
YYYY

**SYSTEM LINE DATA**
LINE # SB EB    LENGTH              Y SHUNT                          Z SERIES
  1     1  7  0.22000E+02  0.00000E+00 0.00000E+00  0.27410E-02 0.56522E-02
  2     2  4  0.25000E+02  0.00000E+00 0.00000E+00  0.11720E-02 0.00000E+00
  3     3  6  0.23000E+02  0.00000E+00 0.00000E+00  0.27505E-02 0.58270E-02
  4     4  5  0.10000E+01  0.00000E+00 0.00000E+00  0.00000E+00 0.62000E-01
  5     5  6  0.90000E+01  0.00000E+00 0.92535E-03  0.15123E-02 0.17958E-02
  6     4  8  0.10000E+01  0.00000E+00 0.00000E+00  0.00000E+00 0.62000E-01
  7     6  9  0.10000E+01  0.00000E+00 0.00000E+00  0.00000E+00 0.46000E-01
  8     7 10  0.10000E+01  0.00000E+00 0.00000E+00  0.00000E+00 0.46000E-01
  9     8 10  0.70000E+01  0.00000E+00 0.92535E-03  0.15123E-02 0.17958E-02

**SYSTEM BUS DATA**
 BUS   TYPE      PG         QG         PD         QD        VMAG       VANG
  1     -1    0.80000    0.50000    0.00000    0.00000    1.00000    0.00000
  2     -3    1.50000    0.30000    0.00000    0.00000    1.00000    0.00000
  3     -2    0.70000    0.20000    0.00000    0.00000    1.00000    0.00000
  4      1    0.00000    0.00000    0.00000    0.00000    1.00000    0.00000
  5      1    0.00000    0.00000    0.70000    0.10000    1.00000    0.00000
  6      1    0.00000    0.00000    0.50000    0.10000    1.00000    0.00000
  7      1    0.00000    0.00000    0.50000    0.30000    1.00000    0.00000
  8      1    0.00000    0.00000    0.30000    0.20000    1.00000    0.00000
  9      2    0.00000    0.00000    0.20000    0.10000    1.00000    0.00000
 10      2    0.00000    0.00000    0.30000    0.30000    1.00000    0.00000

**GENERATOR CONSTRAINTS**
BUS #    PGMAX      PGMIN        QGMAX      QGMIN
  1     0.7500     0.0000
  2     3.0000     0.0000
  3     1.0000     0.0000       2.0000     0.0000
  9                             2.0000     0.0000
 10                             2.0000     0.0000

**SYSTEM CONVERGENCE CRITERIA**
         LOAD FLOW ITERATIONS=  5
         LOAD FLOW TOLERANCE=  0.10000E-03
         ECON DISP ITERATIONS= 50
         ESP1=  0.10000E-02
         ESP2=  0.10000E-02

>>WANT TO CHANGE (LDATA,BDATA,GDATA, OR CONVERGENCE CRITERIA)?
NYYN
```

```
**BUS DATA CHANGES**
>>ENTER(VARIABLE #, BUS #, NEW VALUE THEN 'CR'
          1=BUS TYPE
          2=PG
          3=QG
          4=PD
          5=QD
          6=VMAG
          7=VANG
          8=RETYPE COPY OF ALL BUS DATA
          9=CHANGE TOTAL NUMBER OF BUSES
          10=REENTER ORIGINAL BUS DATA FROM SOS FILE
          11=UPDATE SOS FILE WITH CHANGES
          >>ENTER 0 THEN 'CR' WHEN CHANGES COMPLETED.
1,1,-2
BUS TYPE(BUS #  1)= -2
11
0

**GENERATOR DATA CHANGES**
>>ENTER (VARIABLE #, BUS #, NEW VALUE THEN 'CR')
          1=ALPHA
          2=BETA
          3=GAMMA
          4=PGMAX
          5=PGMIN
          6=QGMAX
          7=QGMIN
          8=RETYPE ALL GENERATOR DATA
          9=UPDATE SOS FILE WITH CHANGES
          >>ENTER 0 THEN 'CR' WHEN CHANGES COMPLETED.
6,1,2.0
QGMAX(BUS #  1)= 0.20000E+01
7,1,0.0
QGMIN(BUS #  1)= 0.00000E+00
9
0

**GENERATING YBUS MATRIX**

>>WANT A COPY OF (YBUS MATRIX, ZBUS MATRIX)?
NN

**SOLVING LOAD FLOW EQUATIONS BY  NR**
BUS #  1  EXCEEDS QGMIN,BUS CHANGED TO TYPE 1
BUS #  3  EXCEEDS QGMIN,BUS CHANGED TO TYPE 1
****NR CONVERGENCE IN  3 ITERATIONS

>>TYPE CODE # FOR DESIRED ACTION
        1=PRINT BUS & GEN SOLUTIONS ONLY.
        2=PRINT LINE  POWERS, BUS & GEN SOLUTIONS,
             AND SYSTEMOPERATING COST.
        3=CREATE SOS FILE OF  SOLUTIONS
        4=CHANGE DATA OR TOLERANCES AND RERUN
        5=CHANGE TYPE PROBLEM OR SOLUTION METHOD,
             RERUN WITH EXISTING DATA
        6=NEW PROBLEM
        7=TERMINATE
 2
```

```
****SYSTEM BUS AND GENERATOR SOLUTIONS****
SYSTEM LOSSES(PU)=  0.1158,  SYSTEM LOAD(PU)=  2.5000
BUS # BTYPE       PG      QG      PD      QD     VMAG    VANG
  1     -1      0.8000  0.0000  0.0000  0.0000  1.0246  0.0667
  2     -3      1.1158 -0.5159  0.0000  0.0000  1.0000  0.0000
  3     -1      0.7000  0.0000  0.0000  0.0000  1.0195  0.0262
  4      1      0.0000  0.0000  0.0000  0.0000  0.9674 -0.0156
  5      1      0.0000  0.0000  0.7000  0.1000  0.9765 -0.0636
  6      1      0.0000  0.0000  0.5000  0.1000  0.9804 -0.0678
  7      1      0.0000  0.0000  0.5000  0.3000  0.9823 -0.0323
  8      1      0.0000  0.0000  0.3000  0.2000  0.9927 -0.0376
  9      2      0.0000  0.5276  0.2000  0.1000  1.0000 -0.0772
 10      2      0.0000  1.2932  0.3000  0.3000  1.0000 -0.0446

****LINE POWER FLOW****
LINE#    SB    EB    REAL POWER     REACTIVE POWER
  1       1     7      0.8000            0.0000
  1       7     1     -0.7632            0.0758
  2       2     4      1.1158           -0.5159
  2       4     2     -1.0715            0.5159
  3       3     6      0.7000            0.0000
  3       6     3     -0.6702            0.0632
  4       4     5      0.7308           -0.1246
  4       5     4     -0.7308            0.1610
  5       5     6      0.0308           -0.2610
  5       6     5     -0.0298            0.2542
  6       4     8      0.3407           -0.3912
  6       8     4     -0.3407            0.4090
  7       6     9      0.2000           -0.4174
  7       9     6     -0.2000            0.4276
  8       7    10      0.2632           -0.3758
  8      10     7     -0.2632            0.3858
  9       8    10      0.0407           -0.6090
  9      10     8     -0.0368            0.6073

>>TYPE CODE # FOR DESIRED ACTION
       1=PRINT BUS & GEN SOLUTIONS ONLY.
       2=PRINT LINE  POWERS, BUS & GEN SOLUTIONS,
            AND SYSTEMOPERATING COST.
       3=CREATE SOS FILE OF  SOLUTIONS
       4=CHANGE DATA OR TOLERANCES AND RERUN
       5=CHANGE TYPE PROBLEM OR SOLUTION METHOD,
            RERUN WITH EXISTING DATA
       6=NEW PROBLEM
       7=TERMINATE
4

>>WANT TO CHANGE (LDATA,BDATA,GDATA, OR CONVERGENCE CRITERIA)?
NYNN

**BUS DATA CHANGES**
>>ENTER(VARIABLE #, BUS #, NEW VALUE THEN 'CR'
         1=BUS TYPE
         2=PG
         3=QG
         4=PD
         5=QD
         6=VMAG
         7=VANG
         8=RETYPE COPY OF ALL BUS DATA
         9=CHANGE TOTAL NUMBER OF BUSES
         10=REENTER ORIGINAL BUS DATA FROM SOS FILE
         11=UPDATE SOS FILE WITH CHANGES
         >>ENTER 0 THEN 'CR' WHEN CHANGES COMPLETED.
```

```
10
6,1,1.05
VMAG(BUS #  1)= 0.10500E+01
6,2,1.05
VMAG(BUS #  2)= 0.10500E+01
6,3,1.05
VMAG(BUS #  3)= 0.10500E+01
8

**SYSTEM BUS DATA**
 BUS   TYPE      PG         QG         PD         QD        VMAG       VANG
  1     -2    0.80000    0.50000    0.00000    0.00000    1.05000    0.00000
  2     -3    1.50000    0.30000    0.00000    0.00000    1.05000    0.00000
  3     -2    0.70000    0.20000    0.00000    0.00000    1.05000    0.00000
  4      1    0.00000    0.00000    0.00000    0.00000    1.00000    0.00000
  5      1    0.00000    0.00000    0.70000    0.10000    1.00000    0.00000
  6      1    0.00000    0.00000    0.50000    0.10000    1.00000    0.00000
  7      1    0.00000    0.00000    0.50000    0.30000    1.00000    0.00000
  8      1    0.00000    0.00000    0.30000    0.20000    1.00000    0.00000
  9      2    0.00000    0.00000    0.20000    0.10000    1.00000    0.00000
 10      2    0.00000    0.00000    0.30000    0.30000    1.00000    0.00000
11
0

>>WANT A COPY OF (YBUS MATRIX, ZBUS MATRIX)?
NN

**SOLVING LOAD FLOW EQUATIONS BY  NR**
****NR CONVERGENCE IN  2 ITERATIONS

>>TYPE CODE # FOR DESIRED ACTION
       1=PRINT BUS & GEN SOLUTIONS ONLY.
       2=PRINT LINE  POWERS, BUS & GEN SOLUTIONS,
            AND SYSTEMOPERATING COST.
       3=CREATE SOS FILE OF  SOLUTIONS
       4=CHANGE DATA OR TOLERANCES AND RERUN
       5=CHANGE TYPE PROBLEM OR SOLUTION METHOD,
            RERUN WITH EXISTING DATA
       6=NEW PROBLEM
       7=TERMINATE
2

****SYSTEM BUS AND GENERATOR SOLUTIONS****
SYSTEM LOSSES(PU)=  0.1055,  SYSTEM LOAD(PU)=  2.5000
BUS # BTYPE       PG      QG      PD      QD      VMAG    VANG
  1    -2       0.8000  0.1534  0.0000  0.0000  1.0500  0.0944
  2    -3       1.1055  0.5551  0.0000  0.0000  1.0500  0.0000
  3    -2       0.7000  0.0736  0.0000  0.0000  1.0500  0.0572
  4     1       0.0000  0.0000  0.0000  0.0000  1.0193  0.0152
  5     1       0.0000  0.0000  0.7000  0.1000  1.0045 -0.0289
  6     1       0.0000  0.0000  0.5000  0.1000  1.0020 -0.0277
  7     1       0.0000  0.0000  0.5000  0.3000  0.9896  0.0075
  8     1       0.0000  0.0000  0.3000  0.2000  1.0015 -0.0052
  9     2       0.0000  0.0567  0.2000  0.1000  1.0000 -0.0368
 10     2       0.0000  0.4367  0.3000  0.3000  1.0000 -0.0048

****LINE POWER FLOW****
LINE#   SB    EB    REAL POWER     REACTIVE POWER
  1      1     7      0.8000            0.1534
  1      7     1     -0.7637           -0.0785
  2      2     4      1.1055            0.5551
  2      4     2     -1.0648           -0.5551
```

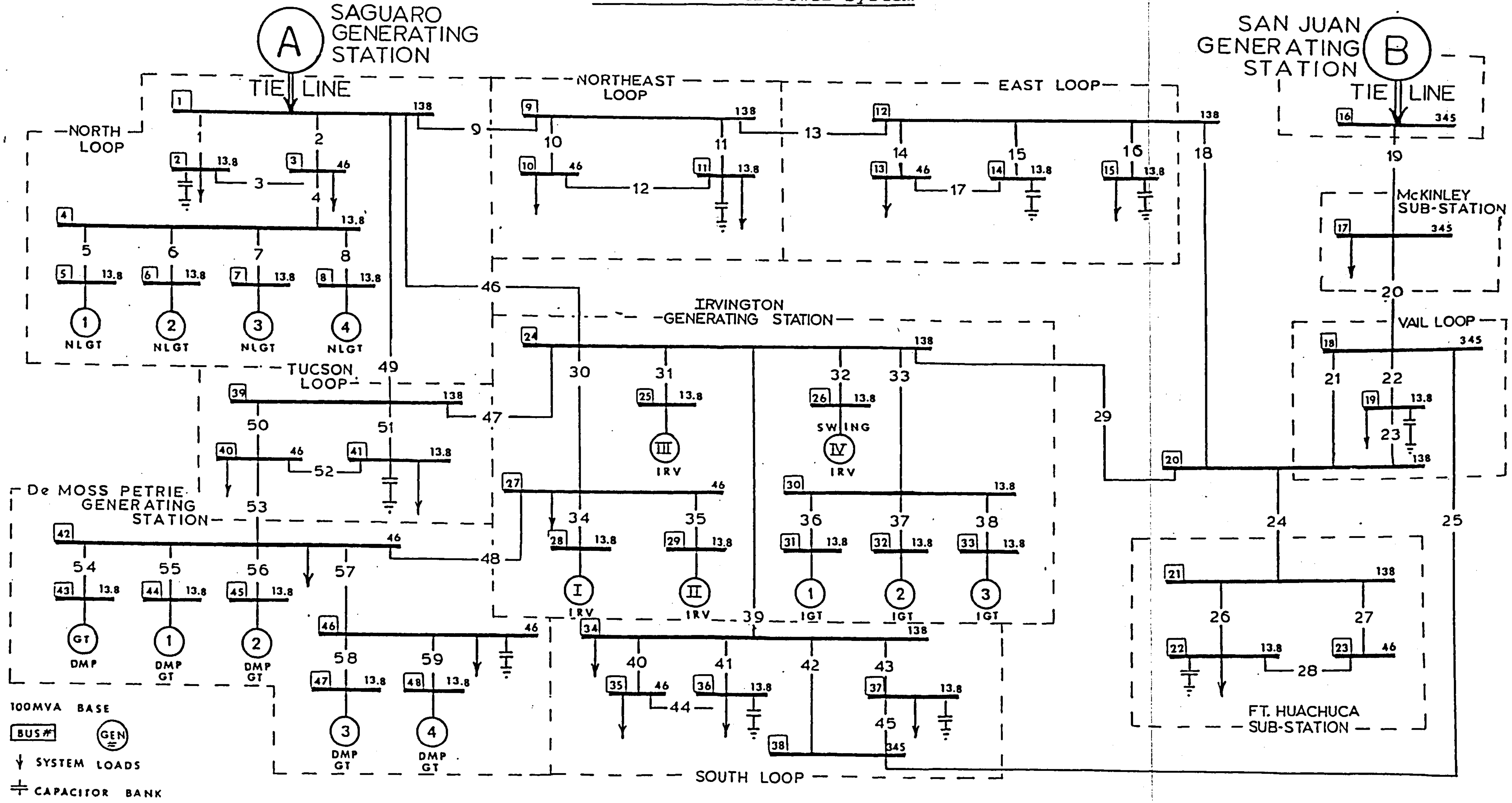
```
    3      3      6      0.7000            0.0736
    3      6      3     -0.6716           -0.0134
    4      4      5      0.7286            0.2593
    4      5      4     -0.7286           -0.2236
    5      5      6      0.0286            0.1237
    5      6      5     -0.0283           -0.1318
    6      4      8      0.3363            0.2959
    6      8      4     -0.3363           -0.2839
    7      6      9      0.2000            0.0452
    7      9      6     -0.2000           -0.0433
    8      7     10      0.2637           -0.2214
    8     10      7     -0.2637            0.2270
    9      8     10      0.0364            0.0839
    9     10      8     -0.0363           -0.0903

>>TYPE CODE # FOR DESIRED ACTION
       1=PRINT BUS & GEN SOLUTIONS ONLY.
       2=PRINT LINE  POWERS, BUS & GEN SOLUTIONS,
           AND SYSTEMOPERATING COST.
       3=CREATE SOS FILE OF  SOLUTIONS
       4=CHANGE DATA OR TOLERANCES AND RERUN
       5=CHANGE TYPE PROBLEM OR SOLUTION METHOD,
           RERUN WITH EXISTING DATA
       6=NEW PROBLEM
       7=TERMINATE
5

>>ENTER TYPE PROBLEM
       0=LOAD FLOW PROBLEM
       1=ECONOMIC DISPATCH PROBLEM
1

>>IS SYSTEM LOSSLESS?
NO

>>ENTER TOTAL NUMBER OF (LINES, & BUSES).
9,10

>>WANT TO TYPE IN NEW(LINE DATA, BUS DATA, OR GENERATOR DATA,
    OR CALCULATE Y/ZBUS MATRIX)?
NNYY

**GENERATOR DATA**
>>ENTER(BUS#,ALPHA,BETA,GAMMA).
1,2.5,5.2,0.02
2,2.0,4.3,0.01
3,2.3,4.8,0.02
**GENERATOR CONSTRAINT DATA**
>>ENTER FOR TYPE  -1 & -3  BUS (BUS #,PGMAX,PGMIN)
        FOR TYPE  -2 BUS       (BUS #,PGMAX,PGMIN,QGMAX,QGMIN)
        FOR TYPE  +2 BUS       (BUS #,QGMAX,QGMIN)
1,0.75,0.0,2.0,0.0
2,2.0,4.3,0.01\10.0,3.4,0.2\3.0,0.0
3,1.0,0.0,2.0,-2.0
9,2.0,-2.0\0.2\0.0
10,2.0,0.0

>>WANT A COPY OF (LDATA,BDATA,GDATA,OR SYSTEM CONVERGENCE CRITERIA)?
YYYYY
```

```
**SYSTEM LINE DATA**
LINE # SB EB    LENGTH              Y SHUNT                        Z SERIES
  1     1  7  0.22000E+02  0.00000E+00 0.00000E+00  0.27410E-02 0.56522E-02
  2     2  4  0.25000E+02  0.00000E+00 0.00000E+00  0.11720E-02 0.00000E+00
  3     3  6  0.23000E+02  0.00000E+00 0.00000E+00  0.27505E-02 0.38270E-02
  4     4  5  0.10000E+01  0.00000E+00 0.00000E+00  0.00000E+00 0.62000E-01
  5     5  6  0.90000E+01  0.00000E+00 0.92535E-03  0.15123E-02 0.17958E-02
  6     4  8  0.10000E+01  0.00000E+00 0.00000E+00  0.00000E+00 0.62000E-01
  7     6  9  0.10000E+01  0.00000E+00 0.00000E+00  0.00000E+00 0.46000E-01
  8     7 10  0.10000E+01  0.00000E+00 0.00000E+00  0.00000E+00 0.46000E-01
  9     8 10  0.70000E+01  0.00000E+00 0.92535E-03  0.15123E-02 0.17958E-02

**SYSTEM BUS DATA**
 BUS    TYPE      PG         QG         PD         QD        VMAG       VANG
  1      -2    0.80000    0.50000    0.00000    0.00000    1.05000    0.00000
  2      -3    1.50000    0.30000    0.00000    0.00000    1.05000    0.00000
  3      -2    0.70000    0.20000    0.00000    0.00000    1.05000    0.00000
  4       1    0.00000    0.00000    0.00000    0.00000    1.00000    0.00000
  5       1    0.00000    0.00000    0.70000    0.10000    1.00000    0.00000
  6       1    0.00000    0.00000    0.50000    0.10000    1.00000    0.00000
  7       1    0.00000    0.00000    0.50000    0.30000    1.00000    0.00000
  8       1    0.00000    0.00000    0.30000    0.20000    1.00000    0.00000
  9       2    0.00000    0.00000    0.20000    0.10000    1.00000    0.00000
 10       2    0.00000    0.00000    0.30000    0.30000    1.00000    0.00000

**SYSTEM GENERATOR DATA**
NUMBER OF GENERATORS =  3
BUS#     ALPHA      BETA       GAMMA
  1    2.50000    5.20000    0.02000
  2    2.00000    4.30000    0.01000
  3    2.30000    4.80000    0.02000

**GENERATOR CONSTRAINTS**
BUS #    PGMAX      PGMIN         QGMAX       QGMIN
  1     0.7500      0.0000       2.0000      0.0000
  2     3.0000      0.0000
  3     1.0000      0.0000       2.0000     -2.0000
  9                              2.0000      0.0000
 10                              2.0000      0.0000

**SYSTEM CONVERGENCE CRITERIA**
         LOAD FLOW ITERATIONS=  5
         LOAD FLOW TOLERANCE=  0.10000E-03
         ECON DISP ITERATIONS= 50
         ESP1=  0.10000E-02
         ESP2=  0.10000E-02

>>WANT TO CHANGE (LDATA,BDATA,GDATA, OR CONVERGENCE CRITERIA)?
NNNN

**GENERATING YBUS MATRIX**
               **CALCULATING ZBUS MATRIX**

>>WANT A COPY OF (YBUS MATRIX, ZBUS MATRIX)?
NN

**SOLVING LOAD FLOW EQUATIONS BY  NR**
****NR CONVERGENCE IN  2 ITERATIONS
    ***OPTIMIZING COST FUNCTIONS***
          LAMBDA DETERMINED IN  15  ITERATIONS
```

```
**SOLVING LOAD FLOW EQUATIONS BY  NR**
****NR CONVERGENCE IN  2 ITERATIONS
    ***OPTIMIZING COST FUNCTIONS***
          LAMBDA DETERMINED IN   5  ITERATIONS

**SOLVING LOAD FLOW EQUATIONS BY  NR**
****NR CONVERGENCE IN  1 ITERATIONS
    ***OPTIMIZING COST FUNCTIONS***
          LAMBDA DETERMINED IN   1  ITERATIONS

**SOLVING LOAD FLOW EQUATIONS BY  NR**
****NR CONVERGENCE IN  1 ITERATIONS
    ***OPTIMIZING COST FUNCTIONS***
          LAMBDA DETERMINED IN   0  ITERATIONS

>>TYPE CODE # FOR DESIRED ACTION
       1=PRINT BUS & GEN SOLUTIONS ONLY.
       2=PRINT LINE  POWERS, BUS & GEN SOLUTIONS,
            AND SYSTEMOPERATING COST.
       3=CREATE SOS FILE OF  SOLUTIONS
       4=CHANGE DATA OR TOLERANCES AND RERUN
       5=CHANGE TYPE PROBLEM OR SOLUTION METHOD,
            RERUN WITH EXISTING DATA
       6=NEW PROBLEM
       7=TERMINATE
2

****SYSTEM BUS AND GENERATOR SOLUTIONS****
     ECON. DISPATCH CONVERGENCE IN  4 ITERATIONS
 ESP1=  0.0010, ESP2=  0.0010,  LAMBDA=  6.9125
SYSTEM LOSSES(PU)=  0.1101,  SYSTEM LOAD(PU)=  2.5000
BUS# BTYPE     IC      ITL      PG      QG      PD      QD      VMAG    VANG
  1  -2     6.3424  0.0825  0.7362  0.1681  0.0000  0.0000  1.0500  0.0804
  2  -3     6.5350  0.0546  1.0523  0.6111  0.0000  0.0000  1.0500  0.0000
  3  -2     6.2532  0.0954  0.8216  0.0340  0.0000  0.0000  1.0500  0.0859
  4   1     0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  1.0208  0.0167
  5   1     0.0000  0.0000  0.0000  0.0000  0.7000  0.1000  1.0032 -0.0207
  6   1     0.0000  0.0000  0.0000  0.0000  0.5000  0.1000  1.0015 -0.0170
  7   1     0.0000  0.0000  0.0000  0.0000  0.5000  0.3000  0.9908  0.0021
  8   1     0.0000  0.0000  0.0000  0.0000  0.3000  0.2000  1.0022 -0.0072
  9   2     0.0000  0.0000  0.0000  0.0693  0.2000  0.1000  1.0000 -0.0262
 10   2     0.0000  0.0000  0.0000  0.3987  0.3000  0.3000  1.0000 -0.0074

****SYSTEM OPERATING COST****
GEN.#  1=    3.25P.U.
GEN.#  2=    4.49P.U.
GEN.#  3=    3.51P.U.
TOTAL SYSTEM OPERATING COST =     11.25P.U.

****LINE POWER FLOW****
LINE#    SB    EB    REAL POWER     REACTIVE POWER
  1      1     7       0.7362            0.1681
  1      7     1      -0.7050           -0.1038
  2      2     4       1.0523            0.6111
  2      4     2      -1.0130           -0.6111
  3      3     6       0.8216            0.0340
  3      6     3      -0.7828            0.0482
  4      4     5       0.6178            0.3012
  4      5     4      -0.6178           -0.2731
  5      5     6      -0.0822            0.1731
  5      6     5       0.0828           -0.1808
  6      4     8       0.3952            0.3100
  6      8     4      -0.3952           -0.2950
  7       6     9      0.2000              0.0326
  7       9     6     -0.2000             -0.0307
  8       7    10      0.2050             -0.1962
  8      10     7     -0.2050              0.2000
  9       8    10      0.0952              0.0950
  9      10     8     -0.0950             -0.1012

>>TYPE CODE # FOR DESIRED ACTION
       1=PRINT BUS & GEN SOLUTIONS ONLY.
       2=PRINT LINE  POWERS, BUS & GEN SOLUTIONS,
            AND SYSTEMOPERATING COST.
       3=CREATE SOS FILE OF  SOLUTIONS
       4=CHANGE DATA OR TOLERANCES AND RERUN
       5=CHANGE TYPE PROBLEM OR SOLUTION METHOD,
            RERUN WITH EXISTING DATA
       6=NEW PROBLEM
       7=TERMINATE
4

>>WANT TO CHANGE (LDATA,BDATA,GDATA, OR CONVERGENCE CRITERIA)?
NNYN

**GENERATOR DATA CHANGES**
>>ENTER (VARIABLE #, BUS #, NEW VALUE THEN 'CR')
         1=ALPHA
         2=BETA
         3=GAMMA
         4=PGMAX
         5=PGMIN
         6=QGMAX
         7=QGMIN
         8=RETYPE ALL GENERATOR DATA
         9=UPDATE SOS FILE WITH CHANGES
         >>ENTER 0 THEN 'CR' WHEN CHANGES COMPLETED.
4,1,.8
PGMAX(BUS #  1)= 0.80000E+00
5,1,.8
PGMIN(BUS #  1)= 0.80000E+00
4,3,.7
PGMAX(BUS #  3)= 0.70000E+00
5,3,.7
PGMIN(BUS #  3)= 0.70000E+00
0

>>WANT A COPY OF (YBUS MATRIX, ZBUS MATRIX)?
NN

**SOLVING LOAD FLOW EQUATIONS BY  NR**
****NR CONVERGENCE IN  0 ITERATIONS
    ***OPTIMIZING COST FUNCTIONS***
          LAMBDA DETERMINED IN  12  ITERATIONS

**SOLVING LOAD FLOW EQUATIONS BY  NR**
****NR CONVERGENCE IN  2 ITERATIONS
    ***OPTIMIZING COST FUNCTIONS***
          LAMBDA DETERMINED IN   2  ITERATIONS

**SOLVING LOAD FLOW EQUATIONS BY  NR**
****NR CONVERGENCE IN  1 ITERATIONS
    ***OPTIMIZING COST FUNCTIONS***
          LAMBDA DETERMINED IN   0  ITERATIONS
```

```
>>TYPE CODE # FOR DESIRED ACTION
       1=PRINT BUS & GEN SOLUTIONS ONLY.
       2=PRINT LINE  POWERS, BUS & GEN SOLUTIONS,
            AND SYSTEMOPERATING COST.
       3=CREATE SOS FILE OF  SOLUTIONS
       4=CHANGE DATA OR TOLERANCES AND RERUN
       5=CHANGE TYPE PROBLEM OR SOLUTION METHOD,
            RERUN WITH EXISTING DATA
       6=NEW PROBLEM
       7=TERMINATE
2

****SYSTEM BUS AND GENERATOR SOLUTIONS****
     ECON. DISPATCH CONVERGENCE IN  3 ITERATIONS
 ESP1=  0.0010, ESP2=  0.0010,  LAMBDA=  7.1875
SYSTEM LOSSES(PU)=  0.1057,  SYSTEM LOAD(PU)=  2.5000
BUS# BTYPE    IC      ITL      PG      QG      PD      QD      VMAG    VANG
  1  -2     6.5313  0.0913  0.8000  0.1534  0.0000  0.0000  1.0500  0.0944
  2  -3     6.7706  0.0580  1.1057  0.5551  0.0000  0.0000  1.0500  0.0000
  3  -2     5.6698  0.0789  0.7000  0.0737  0.0000  0.0000  1.0500  0.0572
  4   1     0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  1.0193  0.0152
  5   1     0.0000  0.0000  0.0000  0.0000  0.7000  0.1000  1.0045 -0.0289
  6   1     0.0000  0.0000  0.0000  0.0000  0.5000  0.1000  1.0020 -0.0277
  7   1     0.0000  0.0000  0.0000  0.0000  0.5000  0.3000  0.9896  0.0075
  8   1     0.0000  0.0000  0.0000  0.0000  0.3000  0.2000  1.0015 -0.0052
  9   2     0.0000  0.0000  0.0000  0.0568  0.2000  0.1000  1.0000 -0.0369
 10   2     0.0000  0.0000  0.0000  0.4368  0.3000  0.3000  1.0000 -0.0048

****SYSTEM OPERATING COST****
GEN.#  1=     3.67P.U.
GEN.#  2=     4.84P.U.
GEN.#  3=     2.79P.U.
TOTAL SYSTEM OPERATING COST =      11.30P.U.

****LINE POWER FLOW****
LINE#    SB    EB    REAL POWER     REACTIVE POWER
  1       1     7      0.8000           0.1534
  1       7     1     -0.7637          -0.0785
  2       2     4      1.1057           0.5551
  2       4     2     -1.0650          -0.5551
  3       3     6      0.7000           0.0737
  3       6     3     -0.6716          -0.0134
  4       4     5      0.7287           0.2593
  4       5     4     -0.7287          -0.2236
  5       5     6      0.0287           0.1236
  5       6     5     -0.0284          -0.1317
  6       4     8      0.3364           0.2958
  6       8     4     -0.3364          -0.2838
  7       6     9      0.2000           0.0451
  7       9     6     -0.2000          -0.0432
  8       7    10      0.2637          -0.2215
  8      10     7     -0.2637           0.2270
  9       8    10      0.0364           0.0838
  9      10     8     -0.0363          -0.0902
```

Problem 3: TGE Power System



```
.RUN EDSP50[3733,547] WITH MODERATE SYSTEM LOAD OF 6.85 pu

               ****ECONOMIC DISPATCH EXECUTIVE PROGRAM****

>>ENTER TYPE PROBLEM
       0=LOAD FLOW PROBLEM
       1=ECONOMIC DISPATCH PROBLEM
0

>>ENTER TOTAL NUMBER OF (LINES, & BUSES).
59,48

>>WANT TO TYPE IN NEW(LINE DATA, BUS DATA, OR GENERATOR DATA,
    OR CALCULATE Y/ZBUS MATRIX)?
NNNN

>>WANT A COPY OF (LDATA,BDATA,GDATA,OR SYSTEM CONVERGENCE CRITERIA)?
YYYY

**SYSTEM LINE DATA**
LINE # SB EB    LENGTH            Y SHUNT                            Z SERIES
  1     1  2  0.10000E+01  0.00000E+00 0.00000E+00  0.00000E+00 0.19100E+00
  2     1  3  0.10000E+01  0.00000E+00 0.00000E+00  0.00000E+00 0.10600E+00
  3     2  3  0.10000E+01  0.00000E+00 0.00000E+00  0.00000E+00 0.62000E-01
  4     3  4  0.10000E+01  0.00000E+00 0.00000E+00  0.00000E+00 0.95500E+00
  5     4  5  0.10000E+01  0.00000E+00 0.00000E+00  0.00000E+00 0.67190E+00
  6     4  6  0.10000E+01  0.00000E+00 0.00000E+00  0.00000E+00 0.67190E+00
  7     4  7  0.10000E+01  0.00000E+00 0.00000E+00  0.00000E+00 0.67190E+00
  8     4  8  0.10000E+01  0.00000E+00 0.00000E+00  0.00000E+00 0.67190E+00
  9     1  9  0.10000E+01  0.00000E+00 0.82000E-02  0.10700E-01 0.62300E-01
 10     9 10  0.10000E+01  0.00000E+00 0.00000E+00  0.00000E+00 0.70400E+00
 11     9 11  0.10000E+01  0.00000E+00 0.00000E+00  0.00000E+00 0.13040E+00
 12    10 11  0.10000E+01  0.00000E+00 0.00000E+00  0.00000E+00 0.45000E-01
 13     9 12  0.10000E+01  0.00000E+00 0.50000E-02  0.47000E-02 0.62300E-01
 14    12 13  0.10000E+01  0.00000E+00 0.00000E+00  0.00000E+00 0.10760E+00
 15    12 14  0.10000E+01  0.00000E+00 0.00000E+00  0.00000E+00 0.19210E+00
 16    12 15  0.10000E+01  0.00000E+00 0.00000E+00  0.00000E+00 0.87500E-01
 17    13 14  0.10000E+01  0.00000E+00 0.00000E+00  0.00000E+00 0.61700E-01
 18    12 20  0.10000E+01  0.00000E+00 0.80000E-02  0.13500E-01 0.78300E-01
 19    16 17  0.10000E+01  0.00000E+00 0.39400E+00  0.40000E-02 0.43400E-01
 20    17 18  0.10000E+01  0.00000E+00 0.14172E+01  0.14100E-01 0.15260E+00
 21    18 20  0.10000E+01  0.00000E+00 0.00000E+00  0.00000E+00 0.16500E-01
 22    18 19  0.10000E+01  0.00000E+00 0.00000E+00  0.00000E+00 0.14940E+00
 23    19 20  0.10000E+01  0.00000E+00 0.00000E+00  0.00000E+00 0.10710E+00
 24    20 21  0.10000E+01  0.00000E+00 0.20000E+00  0.55400E-01 0.21170E+00
 25    18 38  0.10000E+01  0.00000E+00 0.58300E-01  0.60000E-03 0.65000E-02
 26    21 22  0.10000E+01  0.00000E+00 0.00000E+00  0.00000E+00 0.52440E+00
 27    21 23  0.10000E+01  0.00000E+00 0.00000E+00  0.00000E+00 0.27400E+00
 28    22 23  0.10000E+01  0.00000E+00 0.00000E+00  0.00000E+00 0.20500E+00
 29    20 24  0.10000E+01  0.00000E+00 0.92000E-02  0.35800E-02 0.20970E-01
 30    24 27  0.10000E+01  0.00000E+00 0.00000E+00  0.00000E+00 0.12500E+00
 31    24 25  0.10000E+01  0.00000E+00 0.00000E+00  0.00000E+00 0.19320E+00
 32    24 26  0.10000E+01  0.00000E+00 0.00000E+00  0.00000E+00 0.11750E+00
 33    24 30  0.10000E+01  0.00000E+00 0.00000E+00  0.00000E+00 0.24580E+00
 34    27 28  0.10000E+01  0.00000E+00 0.00000E+00  0.00000E+00 0.18650E+00
 35    27 29  0.10000E+01  0.00000E+00 0.00000E+00  0.00000E+00 0.17720E+00
 36    30 31  0.10000E+01  0.00000E+00 0.00000E+00  0.00000E+00 0.43030E+00
 37    30 32  0.10000E+01  0.00000E+00 0.00000E+00  0.00000E+00 0.43030E+00
 38    30 33  0.10000E+01  0.00000E+00 0.00000E+00  0.00000E+00 0.43030E+00
 39    24 34  0.10000E+01  0.00000E+00 0.15700E-01  0.60000E-02 0.31300E-01
 40    34 35  0.10000E+01  0.00000E+00 0.00000E+00  0.00000E+00 0.93800E-01
 41    34 36  0.10000E+01  0.00000E+00 0.00000E+00  0.00000E+00 0.16810E+00
 42    34 38  0.10000E+01  0.00000E+00 0.00000E+00  0.00000E+00 0.16500E-01
 43    34 37  0.10000E+01  0.00000E+00 0.00000E+00  0.00000E+00 0.14940E+00
 44    35 36  0.10000E+01  0.00000E+00 0.00000E+00  0.00000E+00 0.58100E-01
 45    37 38  0.10000E+01  0.00000E+00 0.00000E+00  0.00000E+00 0.10710E+00
 46     1 24  0.10000E+01  0.00000E+00 0.12500E-01  0.17700E-01 0.10580E+00
 47    24 39  0.10000E+01  0.00000E+00 0.52000E-02  0.73000E-02 0.42700E-01
 48    27 42  0.10000E+01  0.00000E+00 0.00000E+00  0.44000E-01 0.26220E+00
 49     1 39  0.10000E+01  0.00000E+00 0.75000E-02  0.10400E-01 0.60700E-01
 50    39 40  0.10000E+01  0.00000E+00 0.00000E+00  0.00000E+00 0.21620E+00
 51    39 41  0.10000E+01  0.00000E+00 0.00000E+00  0.00000E+00 0.12770E+00
 52    40 41  0.10000E+01  0.00000E+00 0.00000E+00  0.00000E+00 0.68100E-01
 53    40 42  0.10000E+01  0.00000E+00 0.00000E+00  0.73000E-02 0.54400E-01
 54    42 43  0.10000E+01  0.00000E+00 0.00000E+00  0.00000E+00 0.33320E+00
 55    42 44  0.10000E+01  0.00000E+00 0.00000E+00  0.00000E+00 0.86280E+00
 56    42 45  0.10000E+01  0.00000E+00 0.00000E+00  0.00000E+00 0.31110E+00
 57    42 46  0.10000E+01  0.00000E+00 0.00000E+00  0.00000E+00 0.16700E+00
 58    46 47  0.10000E+01  0.00000E+00 0.00000E+00  0.00000E+00 0.54310E+00
 59 .  46 48  0.10000E+01  0.00000E+00 0.00000E+00  0.00000E+00 0.84610E+00

**SYSTEM BUS DATA**
 BUS    TYPE      PG         QG         PD         QD        VMAG       VANG
  1      -2    1.10000    0.20000    0.00000    0.00000    1.05000    0.00000
  2       2    0.00000    0.00000    0.06000    0.02000    1.00000    0.00000
  3       1    0.00000    0.00000    0.35000    0.13000    1.00000    0.00000
  4       1    0.00000    0.00000    0.00000    0.00000    1.00000    0.00000
  5      -2    0.15000    0.20000    0.00000    0.00000    1.00000    0.00000
  6      -2    0.15000    0.20000    0.00000    0.00000    1.00000    0.00000
  7      -2    0.25000    0.20000    0.00000    0.00000    1.00000    0.00000
  8      -2    0.25000    0.20000    0.00000    0.00000    1.00000    0.00000
  9       1    0.00000    0.00000    0.00000    0.00000    1.00000    0.00000
 10       1    0.00000    0.00000    0.92000    0.21000    1.00000    0.00000
 11       2    0.00000    0.00000    0.07000    0.01500    1.00000    0.00000
 12       1    0.00000    0.00000    0.00000    0.00000    1.00000    0.00000
 13       1    0.00000    0.00000    0.72000    0.24000    1.00000    0.00000
 14       2    0.00000    0.00000    0.00000    0.00000    1.00000    0.00000
 15       2    0.00000    0.00000    0.37000    0.13000    1.00000    0.00000
 16      -2    1.65000    0.00000    0.00000    0.00000    1.01450    0.00000
 17       1    0.00000    0.00000    0.26000    0.04000    1.00000    0.00000
 18       1    0.00000    0.00000    0.00000    0.00000    1.00000    0.00000
 19       2    0.00000    0.00000    0.02000    0.01800    1.00000    0.00000
 20       1    0.00000    0.00000    0.00000    0.00000    1.00000    0.00000
 21       1    0.00000    0.00000    0.00000    0.00000    1.00000    0.00000
 22       2    0.00000    0.10000    0.25000    0.05000    1.00000    0.00000
 23       1    0.00000    0.00000    0.00000    0.00000    1.00000    0.00000
 24       1    0.00000    0.00000    0.00000    0.00000    1.00000    0.00000
 25      -2    0.50000    0.20000    0.00000    0.00000    1.00000    0.00000
 26      -3    1.20000    0.20000    0.00000    0.00000    1.00000    0.00000
 27       1    0.00000    0.00000    1.12000    0.60000    1.00000    0.00000
 28      -2    0.26000    0.20000    0.00000    0.00000    1.00000    0.00000
 29      -2    0.25000    0.20000    0.00000    0.00000    1.00000    0.00000
 30       1    0.00000    0.00000    0.00000    0.00000    1.00000    0.00000
 31      -1    0.00000    0.00000    0.00000    0.00000    1.00000    0.00000
 32      -1    0.00000    0.00000    0.00000    0.00000    1.00000    0.00000
 33      -1    0.00000    0.00000    0.00000    0.00000    1.00000    0.00000
 34       1    0.00000    0.00000    0.20000    0.35000    1.00000    0.00000
 35       1    0.00000    0.00000    0.31000    0.34000    1.00000    0.00000
 36       2    0.00000    0.00000    0.25000    0.01000    1.00000    0.00000
 37       2    0.00000    0.00000    0.34000    0.25000    1.00000    0.00000
 38       1    0.00000    0.00000    0.00000    0.00000    1.00000    0.00000
 39       1    0.00000    0.00000    0.00000    0.00000    1.00000    0.00000
 40       1    0.00000    0.00000    0.42000    0.15000    1.00000    0.00000
 41       2    0.00000    0.20000    0.36000    0.12000    1.00000    0.00000
 42       1    0.00000    0.00000    0.61000    0.21000    1.00000    0.00000
 43      -2    0.48100    0.20000    0.00000    0.00000    1.00000    0.00000
 44      -2    0.14000    0.00000    0.00000    0.00000    1.00000    0.00000
```

```
  45     -2   0.13000   0.00000   0.00000   0.00000   1.00000   0.00000
  46      2   0.00000   0.00000   0.22000   0.07000   1.00000   0.00000
  47     -2   0.24500   0.10000   0.00000   0.00000   1.00000   0.00000
  48     -2   0.46000   0.15000   0.00000   0.00000   1.00000   0.00000

**GENERATOR CONSTRAINTS**
BUS #    PGMAX     PGMIN        QGMAX     QGMIN
  1      1.1000     0.0000      0.2000   -2.0000
  2                             2.0000   -2.0000
  5      0.2510     0.0000      2.0000   -2.0000
  6      0.2510     0.0000      2.0000   -2.0000
  7      0.2510     0.0000      2.0000   -2.0000
  8      0.2510     0.0000      2.0000   -2.0000
 11                             0.8000   -2.0000
 14                             0.4000   -2.0000
 15                             0.4000   -2.0000
 16      1.6500     0.0000      2.0000   -2.0000
 19                             0.4500   -2.0000
 22                             0.2400   -2.0000
 25      1.0900     0.0000      2.0000   -2.0000
 26      1.5900     0.0000
 28      0.7700     0.0000      2.0000   -2.0000
 29      0.7900     0.0000      2.0000   -2.0000
 31      0.2490     0.0000
 32      0.2490     0.0000
 33      0.2490     0.0000
 36                             0.4000   -2.0000
 37                             0.4300   -2.0000
 41                             0.4000   -2.0000
 43      0.4810     0.0000      0.4500   -2.0000
 44      0.1480     0.0000      0.1500   -2.0000
 45      0.1320     0.0000      0.1400   -2.0000
 46                             0.3600   -2.0000
 47      0.2450     0.0000      0.2500   -2.0000
 48      0.4680     0.0000      0.4500   -2.0000

**SYSTEM CONVERGENCE CRITERIA**
         LOAD FLOW ITERATIONS=  5
         LOAD FLOW TOLERANCE=  0.10000E-03
         ECON DISP ITERATIONS= 50
         ESP1=  0.10000E-02
         ESP2=  0.10000E-02

>>WANT TO CHANGE (LDATA,BDATA,GDATA, OR CONVERGENCE CRITERIA)?
NNNN\NNNN\YYYY

** LINE DATA CHANGES**
>>ENTER (VARIABLE #, LINE #, NEW VALUE THEN 'CR')
         1=STARTING BUS #
         2=ENDING BUS #
         3=LINE LENGTH
         4=Y SHUNT G
         5=Y SHUNT B
         6=Z SERIES R
         7=Z SERIES X
         8=RETYPE COPY OF ALL LINE DATA
         9=CHANGE TOTAL NUMBER OF LINES
         10=UPDATE SOS FILE WITH CHANGES
         >>ENTER 0 THEN'CR' WHEN CHANGES COMPLETED
0
```

```
**BUS DATA CHANGES**
>>ENTER(VARIABLE #, BUS #, NEW VALUE THEN 'CR'
         1=BUS TYPE
         2=PG
         3=QG
         4=PD
         5=QD
         6=VMAG
         7=VANG
         8=RETYPE COPY OF ALL BUS DATA
         9=CHANGE TOTAL NUMBER OF BUSES
         10=REENTER ORIGINAL BUS DATA FROM SOS FILE
         11=UPDATE SOS FILE WITH CHANGES
         >>ENTER 0 THEN 'CR' WHEN CHANGES COMPLETED.
0

**GENERATOR DATA CHANGES**
>>ENTER (VARIABLE #, BUS #, NEW VALUE THEN 'CR')
         1=ALPHA
         2=BETA
         3=GAMMA
         4=PGMAX
         5=PGMIN
         6=QGMAX
         7=QGMIN
         8=RETYPE ALL GENERATOR DATA
         9=UPDATE SOS FILE WITH CHANGES
         >>ENTER 0 THEN 'CR' WHEN CHANGES COMPLETED.
0

**CONVERGENCE CRITERIA CHANGES**
>>ENTER(VARIABLE #,NEW VALUE THEN 'CR')
         1=LOAD FLOW ITERATIONS
         2=LOAD FLOW TOLERANCE
         3=ECON DISP ITERATIONS
         4=ESP1
         5=ESP2
         >>ENTER 0 THEN 'CR' WHEN CHANGES COMPLETED.
0
  **GENERATING YBUS MATRIX**

  >>WANT A COPY OF (YBUS MATRIX, ZBUS MATRIX)?
  NN

  **SOLVING LOAD FLOW EQUATIONS BY  NR**
  BUS #  1  EXCEEDS QGMAX,BUS CHANGED TO TYPE 1
  ****NR CONVERGENCE IN  4 ITERATIONS

  >>TYPE CODE # FOR DESIRED ACTION
         1=PRINT BUS & GEN SOLUTIONS ONLY.
         2=PRINT LINE  POWERS, BUS & GEN SOLUTIONS,
              AND SYSTEM OPERATING COST.
         3=CREATE SOS FILE OF  SOLUTIONS
         4=CHANGE DATA OR TOLERANCES AND RERUN
         5=CHANGE TYPE PROBLEM OR SOLUTION METHOD,
              RERUN WITH EXISTING DATA
         6=NEW PROBLEM
         7=TERMINATE
  2
```

```
****SYSTEM BUS AND GENERATOR SOLUTIONS****
SYSTEM LOSSES(PU)=  0.0746,  SYSTEM LOAD(PU)=  6.8500
BUS # BTYPE       PG      QG       PD      QD      VMAG    VANG
  1    -1       1.1000  0.2000  0.0000  0.0000  0.9981 -0.0997
  2     2       0.0000  0.4621  0.0600  0.0200  1.0000 -0.0794
  3     1       0.0000  0.0000  0.3500  0.1300  0.9733 -0.0688
  4     1       0.0000  0.0000  0.0000  0.0000  0.9156  0.9612
  5    -2       0.1500  0.1339  0.0000  0.0000  1.0000  1.0715
  6    -2       0.1500  0.1339  0.0000  0.0000  1.0000  1.0715
  7    -2       0.2500  0.1488  0.0000  0.0000  1.0000  1.1457
  8    -2       0.2500  0.1488  0.0000  0.0000  1.0000  1.1457
  9     1       0.0000  0.0000  0.0000  0.0000  0.9883 -0.1762
 10     1       0.0000  0.0000  0.9200  0.2100  0.9892 -0.3148
 11     2       0.0000  0.3979  0.0700  0.0150  1.0000 -0.2817
 12     1       0.0000  0.0000  0.0000  0.0000  0.9909 -0.1893
 13     1       0.0000  0.0000  0.7200  0.2400  0.9865 -0.2448
 14     2       0.0000  0.2717  0.0000  0.0000  1.0000 -0.2313
 15     2       0.0000  0.2401  0.3700  0.1300  1.0000 -0.2220
 16    -2       1.6500 -0.7939  0.0000  0.0000  1.0145  0.1620
 17     1       0.0000  0.0000  0.2600  0.0400  1.0359  0.0915
 18     1       0.0000  0.0000  0.0000  0.0000  1.0061 -0.1087
 19     2       0.0000 -0.0486  0.0200  0.0180  1.0000 -0.1159
 20     1       0.0000  0.0000  0.0000  0.0000  1.0028 -0.1189
 21     1       0.0000  0.0000  0.0000  0.0000  1.0038 -0.1758
 22     2       0.0000  0.0427  0.2500  0.0500  1.0000 -0.2381
 23     1       0.0000  0.0000  0.0000  0.0000  1.0011 -0.2114
 24     1       0.0000  0.0000  0.0000  0.0000  0.9980 -0.1072
 25    -2       0.5000  0.0346  0.0000  0.0000  1.0000 -0.0102
 26    -3       0.9086  0.0657  0.0000  0.0000  1.0000  0.0000
 27     1       0.0000  0.0000  1.1200  0.6000  0.9666 -0.1617
 28    -2       0.2600  0.1859  0.0000  0.0000  1.0000 -0.1115
 29    -2       0.2500  0.1945  0.0000  0.0000  1.0000 -0.1159
 30     1       0.0000  0.0000  0.0000  0.0000  0.9980 -0.1072
 31    -1       0.0000  0.0000  0.0000  0.0000  0.9980 -0.1072
 32    -1       0.0000  0.0000  0.0000  0.0000  0.9980 -0.1072
 33    -1       0.0000  0.0000  0.0000  0.0000  0.9980 -0.1072
 34     1       0.0000  0.0000  0.2000  0.3500  0.9959 -0.1204
 35     1       0.0000  0.0000  0.3100  0.3400  0.9858 -0.1537
 36     2       0.0000  0.2821  0.2500  0.0100  1.0000 -0.1560
 37     2       0.0000  0.2542  0.3400  0.2500  1.0000 -0.1372
 38     1       0.0000  0.0000  0.0000  0.0000  1.0029 -0.1129
 39     1       0.0000  0.0000  0.0000  0.0000  0.9961 -0.1128
 40     1       0.0000  0.0000  0.4200  0.1500  0.9855 -0.1357
 41     2       0.0000  0.3674  0.3600  0.1200  1.0000 -0.1439
 42     1       0.0000  0.0000  0.6100  0.2100  0.9832 -0.1106
 43    -2       0.4810  0.0899  0.0000  0.0000  1.0000  0.0532
 44    -2       0.1400  0.0281  0.0000  0.0000  1.0000  0.0126
 45    -2       0.1300  0.0567  0.0000  0.0000  1.0000 -0.0694
 46     2       0.0000  0.3003  0.2200  0.0700  1.0000 -0.0281
 47    -2       0.2450  0.0164  0.0000  0.0000  1.0000  0.1053
 48    -2       0.4600  0.0932  0.0000  0.0000  1.0000  0.3717

****LINE POWER FLOW****
LINE#    SB     EB     REAL POWER     REACTIVE POWER
  1       1      2     -0.1064           -0.0086
  1       2      1      0.1064            0.0108
  2       1      3     -0.2836            0.2381
  2       3      1      0.2836           -0.2236
  3       2      3     -0.1664            0.4312
  3       3      2      0.1664           -0.4180
  4       3      4     -0.8000            0.5116
  4       4      3      0.8000            0.3974
```

5	4	5	-0.1500	-0.1068
5	5	4	0.1500	0.1339
6	4	6	-0.1500	-0.1068
6	6	4	0.1500	0.1339
7	4	7	-0.2500	-0.0919
7	7	4	0.2500	0.1488
8	4	8	-0.2500	-0.0919
8	8	4	0.2500	0.1488
9	1	9	1.2084	-0.0082
9	9	1	-1.1927	0.0914
10	9	10	0.1919	0.0120
10	10	9	-0.1919	0.0146
11	9	11	0.7981	-0.0463
11	11	9	-0.7981	0.1316
12	10	11	-0.7281	-0.2246
12	11	10	0.7281	0.2513
13	9	12	0.2027	-0.0571
13	12	9	-0.2025	0.0550
14	12	13	0.5037	0.0542
14	13	12	-0.5037	-0.0261
15	12	14	0.2163	-0.0424
15	14	12	-0.2163	0.0519
16	12	15	0.3700	-0.0971
16	15	12	-0.3700	0.1101
17	13	14	-0.2163	-0.2139
17	14	13	0.2163	0.2198
18	12	20	-0.8875	0.0303
18	20	12	0.8984	0.0247
19	16	17	1.6500	-0.7939
19	17	16	-1.6381	0.5093
20	17	18	1.3781	-0.5493
20	18	17	-1.3525	-0.6519
21	18	20	0.6206	0.2064
21	20	18	-0.6206	-0.1994
22	18	19	0.0481	0.0413
22	19	18	-0.0481	-0.0407
23	19	20	0.0281	-0.0259
23	20	19	-0.0281	0.0260
24	20	21	0.2538	-0.1640
24	21	20	-0.2500	-0.0229
25	18	38	0.6838	0.4042
25	38	18	-0.6835	-0.4588
26	21	22	0.1193	0.0109
26	22	21	-0.1193	-0.0035
27	21	23	0.1307	0.0120
27	23	21	-0.1307	-0.0073
28	22	23	-0.1307	-0.0038
28	23	22	0.1307	0.0073
29	20	24	-0.5035	0.3127
29	24	20	0.5047	-0.3145
30	24	27	0.4206	0.2626
30	27	24	-0.4206	-0.2317
31	24	25	-0.5000	0.0139
31	25	24	0.5000	0.0346
32	24	26	-0.9086	0.0318
32	26	24	0.9086	0.0657
33	24	30	-0.0000	0.0000
33	30	24	0.0000	0.0000
34	27	28	-0.2600	-0.1668
34	28	27	0.2600	0.1859
35	27	29	-0.2500	-0.1767
35	29	27	0.2500	0.1945
36	30	31	-0.0000	0.0000
36	31	30	0.0000	0.0000

```
 37      30     32      -0.0000                 0.0000
 37      32     30       0.0000                 0.0000
 38      30     33      -0.0000                 0.0000
 38      33     30       0.0000                 0.0000
 39      24     34       0.4176                -0.0173
 39      34     24      -0.4165                 0.0072
 40      34     35       0.3489                 0.1124
 40      35     34      -0.3489                -0.0997
 41      34     36       0.2111                -0.0207
 41      36     34      -0.2111                 0.0283
 42      34     38      -0.4555                -0.4224
 42      38     34       0.4555                 0.4289
 43      34     37       0.1121                -0.0265
 43      37     34      -0.1121                 0.0285
 44      35     36       0.0389                -0.2403
 44      36     35      -0.0389                 0.2438
 45      37     38      -0.2279                -0.0244
 45      38     37       0.2279                 0.0300
 46       1     24       0.0684                -0.0161
 46      24      1      -0.0683                 0.0042
 47      24     39       0.1340                 0.0194
 47      39     24      -0.1338                -0.0238
 48      27     42      -0.1894                -0.0248
 48      42     27       0.1911                 0.0350
 49       1     39       0.2132                -0.0053
 49      39      1      -0.2128                 0.0006
 50      39     40       0.1042                 0.0499
 50      40     39      -0.1042                -0.0470
 51      39     41       0.2424                -0.0267
 51      41     39      -0.2424                 0.0343
 52      40     41       0.1176                -0.2091
 52      41     40      -0.1176                 0.2131
 53      40     42      -0.4334                 0.1061
 53      42     40       0.4349                -0.0949
 54      42     43      -0.4810                -0.0101
 54      43     42       0.4810                 0.0899
 55      42     44      -0.1400                -0.0105
 55      44     42       0.1400                 0.0281
 56      42     45      -0.1300                -0.0505
 56      45     42       0.1300                 0.0567
 57      42     46      -0.4850                -0.0790
 57      46     42       0.4850                 0.1207
 58      46     47      -0.2450                 0.0164
 58      47     46       0.2450                 0.0164
 59      46     48      -0.4600                 0.0932
 59      48     46       0.4600                 0.0932

>>TYPE CODE # FOR DESIRED ACTION
       1=PRINT BUS & GEN SOLUTIONS ONLY.
       2=PRINT LINE  POWERS, BUS & GEN SOLUTIONS,
            AND SYSTEM OPERATING COST.
       3=CREATE SOS FILE OF  SOLUTIONS
       4=CHANGE DATA OR TOLERANCES AND RERUN
       5=CHANGE TYPE PROBLEM OR SOLUTION METHOD,
            RERUN WITH EXISTING DATA
       6=NEW PROBLEM
       7=TERMINATE
5
              ****ECONOMIC DISPATCH EXECUTIVE PROGRAM****

>>ENTER TYPE PROBLEM
       0=LOAD FLOW PROBLEM
       1=ECONOMIC DISPATCH PROBLEM
1
```

```
>>IS SYSTEM LOSSLESS?
NO

>>ENTER TOTAL NUMBER OF (LINES, & BUSES).
59,48

>>WANT TO TYPE IN NEW(LINE DATA, BUS DATA, OR GENERATOR DATA,
    OR CALCULATE Y/ZBUS MATRIX)?
NNNY

>>WANT A COPY OF (LDATA,BDATA,GDATA,OR SYSTEM CONVERGENCE CRITERIA)?
NNYN

**SYSTEM GENERATOR DATA**
NUMBER OF GENERATORS = 18
BUS#     ALPHA     BETA       GAMMA
  1    0.00000   0.00000   0.00000
  5    9.56170   5.52200  -2.12000
  6    9.56170   5.52200  -2.12000
  7    9.56170   5.52200  -2.12000
  8    9.56170   5.52200  -2.12000
 16    0.00000   0.00000   0.00000
 25    8.70820   1.24000   0.08900
 26    8.39970   0.42050   0.13000
 28    8.80940   1.00100   0.44100
 29    7.98410   2.78000   0.17000
 31    9.56170   5.52200  -2.12000
 32    9.56170   5.52200  -2.12000
 33    9.56170   5.52200  -2.12000
 43    9.46170   4.52200  -2.12000
 44    9.20980  13.87000  19.33000
 45    9.20980  12.87000  18.33000
 47    9.56170   5.52200  -2.12000
 48    9.51170   5.22200  -2.12000

**GENERATOR CONSTRAINTS**
BUS #     PGMAX      PGMIN        QGMAX       QGMIN
  1      1.1000     0.0000      0.2000     -2.0000
  2                             2.0000     -2.0000
  5      0.2510     0.0000      2.0000     -2.0000
  6      0.2510     0.0000      2.0000     -2.0000
  7      0.2510     0.0000      2.0000     -2.0000
  8      0.2510     0.0000      2.0000     -2.0000
 11                             0.8000     -2.0000
 14                             0.4000     -2.0000
 15                             0.4000     -2.0000
 16      1.6500     0.0000      2.0000     -2.0000
 19                             0.4500     -2.0000
 22                             0.2400     -2.0000
 25      1.0900     0.0000      2.0000     -2.0000
 26      1.5900     0.0000
 28      0.7700     0.0000      2.0000     -2.0000
 29      0.7900     0.0000      2.0000     -2.0000
 31      0.2490     0.0000
 32      0.2490     0.0000
 33      0.2490     0.0000
 36                             0.4000     -2.0000
 37                             0.4300     -2.0000
 41                             0.4000     -2.0000
 43      0.4810     0.0000      0.4500     -2.0000
 44      0.1480     0.0000      0.1500     -2.0000
 45      0.1320     0.0000      0.1400     -2.0000
 46                             0.3600     -2.0000
 47      0.2450      0.0000      0.2500    -2.0000
 48      0.4680      0.0000      0.4500    -2.0000

>>WANT TO CHANGE (LDATA,BDATA,GDATA, OR CONVERGENCE CRITERIA)?
NNNN

**GENERATING YBUS MATRIX**
              **CALCULATING ZBUS MATRIX**
[Exceeding quota on STDN]

[Exceeding quota on STDN]


>>WANT A COPY OF (YBUS MATRIX, ZBUS MATRIX)?
NN

**SOLVING LOAD FLOW EQUATIONS BY  NR**
BUS #  1  EXCEEDS QGMAX,BUS CHANGED TO TYPE 1
****NR CONVERGENCE IN  3 ITERATIONS
    ***OPTIMIZING COST FUNCTIONS***
          LAMBDA DETERMINED IN  20  ITERATIONS

**SOLVING LOAD FLOW EQUATIONS BY  NR**
BUS # 41  EXCEEDS QGMAX,BUS CHANGED TO TYPE 1
BUS #  1  EXCEEDS QGMAX,BUS CHANGED TO TYPE 1
****NR CONVERGENCE IN  4 ITERATIONS
    ***OPTIMIZING COST FUNCTIONS***
          LAMBDA DETERMINED IN  16  ITERATIONS

**SOLVING LOAD FLOW EQUATIONS BY  NR**
BUS #  1  EXCEEDS QGMAX,BUS CHANGED TO TYPE 1
BUS # 41  EXCEEDS QGMAX,BUS CHANGED TO TYPE 1
****NR CONVERGENCE IN  3 ITERATIONS
    ***OPTIMIZING COST FUNCTIONS***
          LAMBDA DETERMINED IN  22  ITERATIONS

**SOLVING LOAD FLOW EQUATIONS BY  NR**
BUS #  1  EXCEEDS QGMAX,BUS CHANGED TO TYPE 1
BUS # 41  EXCEEDS QGMAX,BUS CHANGED TO TYPE 1
****NR CONVERGENCE IN  2 ITERATIONS
    ***OPTIMIZING COST FUNCTIONS***
          LAMBDA DETERMINED IN  21  ITERATIONS

**SOLVING LOAD FLOW EQUATIONS BY  NR**
BUS #  1  EXCEEDS QGMAX,BUS CHANGED TO TYPE 1
BUS # 41  EXCEEDS QGMAX,BUS CHANGED TO TYPE 1
****NR CONVERGENCE IN  2 ITERATIONS
    ***OPTIMIZING COST FUNCTIONS***
          LAMBDA DETERMINED IN  20  ITERATIONS

>>TYPE CODE # FOR DESIRED ACTION
       1=PRINT BUS & GEN SOLUTIONS ONLY.
       2=PRINT LINE  POWERS, BUS & GEN SOLUTIONS,
            AND SYSTEM OPERATING COST.
       3=CREATE SOS FILE OF  SOLUTIONS
       4=CHANGE DATA OR TOLERANCES AND RERUN
       5=CHANGE TYPE PROBLEM OR SOLUTION METHOD,
            RERUN WITH EXISTING DATA
       6=NEW PROBLEM
       7=TERMINATE
2
```

```
****SYSTEM BUS AND GENERATOR SOLUTIONS****
     ECON. DISPATCH CONVERGENCE IN  5 ITERATIONS
 ESP1=  0.0010, ESP2=  0.0010,  LAMBDA= 10.3902
SYSTEM LOSSES(PU)=  0.0889,  SYSTEM LOAD(PU)=  6.8500
BUS# BTYPE    IC       ITL       PG       QG       PD       QD      VMAG     VANG
  1  -1     0.0000   0.0701   1.1000   0.2000   0.0000   0.0000   0.9990 -0.2242
  2   2     0.0000   0.0000   0.0000   0.1162   0.0600   0.0200   1.0000 -0.2445
  3   1     0.0000   0.0000   0.0000   0.0000   0.3500   0.1300   0.9944 -0.2474
  4   1     0.0000   0.0000   0.0000   0.0000   0.0000   0.0000   0.9986 -0.1639
  5  -2     9.6795   0.0684   0.0217   0.0023   0.0000   0.0000   1.0000 -0.1493
  6  -2     9.6795   0.0684   0.0217   0.0023   0.0000   0.0000   1.0000 -0.1493
  7  -2     9.6795   0.0684   0.0217   0.0023   0.0000   0.0000   1.0000 -0.1493
  8  -2     9.6795   0.0684   0.0217   0.0023   0.0000   0.0000   1.0000 -0.1493
  9   1     0.0000   0.0000   0.0000   0.0000   0.0000   0.0000   0.9893 -0.2882
 10   1     0.0000   0.0000   0.0000   0.0000   0.9200   0.2100   0.9893 -0.4267
 11   2     0.0000   0.0000   0.0000   0.3889   0.0700   0.0150   1.0000 -0.3936
 12   1     0.0000   0.0000   0.0000   0.0000   0.0000   0.0000   0.9905 -0.2893
 13   1     0.0000   0.0000   0.0000   0.0000   0.7200   0.2400   0.9864 -0.3448
 14   2     0.0000   0.0000   0.0000   0.2762   0.0000   0.0000   1.0000 -0.3312
 15   2     0.0000   0.0000   0.0000   0.2447   0.3700   0.1300   1.0000 -0.3220
 16  -2     0.0000  -0.0204   1.6500  -0.7924   0.0000   0.0000   1.0145  0.0783
 17   1     0.0000   0.0000   0.0000   0.0000   0.2600   0.0400   1.0358  0.0078
 18   1     0.0000   0.0000   0.0000   0.0000   0.0000   0.0000   1.0058 -0.1925
 19   2     0.0000   0.0000   0.0000  -0.0436   0.0200   0.0180   1.0000 -0.2000
 20   1     0.0000   0.0000   0.0000   0.0000   0.0000   0.0000   1.0024 -0.2033
 21   1     0.0000   0.0000   0.0000   0.0000   0.0000   0.0000   1.0036 -0.2603
 22   2     0.0000   0.0000   0.0000   0.0435   0.2500   0.0500   1.0000 -0.3227
 23   1     0.0000   0.0000   0.0000   0.0000   0.0000   0.0000   1.0011 -0.2959
 24   1     0.0000   0.0000   0.0000   0.0000   0.0000   0.0000   0.9978 -0.1885
 25  -2     9.8304   0.0539   0.8523   0.0824   0.0000   0.0000   1.0000 -0.0227
 26  -3     9.3969   0.0533   1.5909   0.1694   0.0000   0.0000   1.0000  0.0000
 27   1     0.0000   0.0000   0.0000   0.0000   1.1200   0.6000   0.9665 -0.2153
 28  -2     9.7449   0.0621   0.7111   0.2285   0.0000   0.0000   1.0000 -0.0777
 29  -2     9.7380   0.0628   0.6081   0.2229   0.0000   0.0000   1.0000 -0.1036
 30   1     0.0000   0.0000   0.0000   0.0000   0.0000   0.0000   0.9966 -0.1560
 31  -1     9.7998   0.0568   0.0437   0.0000   0.0000   0.0000   0.9964 -0.1371
 32  -1     9.7998   0.0568   0.0437   0.0000   0.0000   0.0000   0.9964 -0.1371
 33  -1     9.7998   0.0568   0.0437   0.0000   0.0000   0.0000   0.9964 -0.1371
 34   1     0.0000   0.0000   0.0000   0.0000   0.2000   0.3500   0.9956 -0.2031
 35   1     0.0000   0.0000   0.0000   0.0000   0.3100   0.3400   0.9857 -0.2365
 36   2     0.0000   0.0000   0.0000   0.2857   0.2500   0.0100   1.0000 -0.2388
 37   2     0.0000   0.0000   0.0000   0.2584   0.3400   0.2500   1.0000 -0.2204
 38   1     0.0000   0.0000   0.0000   0.0000   0.0000   0.0000   1.0027 -0.1963
 39   1     0.0000   0.0000   0.0000   0.0000   0.0000   0.0000   0.9928 -0.2285
 40   1     0.0000   0.0000   0.0000   0.0000   0.4200   0.1500   0.9843 -0.3203
 41   1     0.0000   0.0000   0.0000   0.4000   0.3600   0.1200   0.9986 -0.3045
 42   1     0.0000   0.0000   0.0000   0.0000   0.6100   0.2100   0.9787 -0.3324
 43  -2     9.7315   0.0634   0.0614   0.0647   0.0000   0.0000   1.0000 -0.3115
 44  -2     9.7331   0.0632   0.0359   0.0253   0.0000   0.0000   1.0000 -0.3007
 45  -2     9.7303   0.0635   0.0383   0.0688   0.0000   0.0000   1.0000 -0.3202
 46   2     0.0000   0.0000   0.0000   0.2006   0.2200   0.0700   1.0000 -0.3575
 47  -2     9.7275   0.0638   0.0303   0.0003   0.0000   0.0000   1.0000 -0.3410
 48  -2     9.7301   0.0635   0.0425   0.0008   0.0000   0.0000   1.0000 -0.3215

****SYSTEM OPERATING COST****
GEN.#  1=     0.00P.U.
GEN.#  5=     0.21P.U.
GEN.#  6=     0.21P.U.
GEN.#  7=     0.21P.U.
GEN.#  8=     0.21P.U.
GEN.# 16=     0.00P.U.
GEN.# 25=     7.89P.U.
GEN.# 26=    14.07P.U.
GEN.# 28=    6.57P.U.
GEN.# 29=    5.38P.U.
GEN.# 31=    0.42P.U.
GEN.# 32=    0.42P.U.
GEN.# 33=    0.42P.U.
GEN.# 43=    0.59P.U.
GEN.# 44=    0.34P.U.
GEN.# 45=    0.36P.U.
GEN.# 47=    0.29P.U.
GEN.# 48=    0.41P.U.
TOTAL SYSTEM OPERATING COST =     38.01P.U.

****LINE POWER FLOW****
LINE#    SB     EB    REAL POWER     REACTIVE POWER
   1      1      2      0.1061           -0.0042
   1      2      1     -0.1061            0.0064
   2      1      3      0.2171            0.0454
   2      3      1     -0.2171           -0.0402
   3      2      3      0.0461            0.0898
   3      3      2     -0.0461           -0.0892
   4      3      4     -0.0868           -0.0007
   4      4      3      0.0868            0.0079
   5      4      5     -0.0217           -0.0020
   5      5      4      0.0217            0.0023
   6      4      6     -0.0217           -0.0020
   6      6      4      0.0217            0.0023
   7      4      7     -0.0217           -0.0020
   7      7      4      0.0217            0.0023
   8      4      8     -0.0217           -0.0020
   8      8      4      0.0217            0.0023
   9      1      9      1.0164            0.0087
   9      9      1     -1.0053            0.0477
  10      9     10      0.1919            0.0134
  10     10      9     -0.1919            0.0133
  11      9     11      0.7981           -0.0389
  11     11      9     -0.7981            0.1240
  12     10     11     -0.7281           -0.2233
  12     11     10      0.7281            0.2499
  13      9     12      0.0154           -0.0221
  13     12      9     -0.0153            0.0173
  14     12     13      0.5037            0.0518
  14     13     12     -0.5037           -0.0237
  15     12     14      0.2163           -0.0445
  15     14     12     -0.2163            0.0540
  16     12     15      0.3700           -0.1016
  16     15     12     -0.3700            0.1147
  17     13     14     -0.2163           -0.2163
  17     14     13      0.2163            0.2222
  18     12     20     -1.0747            0.0769
  18     20     12      1.0906            0.0078
  19     16     17      1.6500           -0.7924
  19     17     16     -1.6381            0.5077
  20     17     18      1.3781           -0.5477
  20     18     17     -1.3525           -0.6529
  21     18     20      0.6631            0.2099
  21     20     18     -0.6631           -0.2020
  22     18     19      0.0509            0.0395
  22     19     18     -0.0509           -0.0388
  23     19     20      0.0309           -0.0228
  23     20     19     -0.0309            0.0230
  24     20     21      0.2538           -0.1646
  24     21     20     -0.2500           -0.0222
  25     18     38      0.6386            0.4035
  25     38     18     -0.6382           -0.4585
```

26	21	22	0.1193	0.0106
26	22	21	-0.1193	-0.0031
27	21	23	0.1307	0.0116
27	23	21	-0.1307	-0.0069
28	22	23	-0.1307	-0.0034
28	23	22	0.1307	0.0069
29	20	24	-0.6505	0.3358
29	24	20	0.6524	-0.3338
30	24	27	0.2074	0.2521
30	27	24	-0.2074	-0.2387
31	24	25	-0.8523	0.0593
31	25	24	0.8523	0.0824
32	24	26	-1.5909	0.1313
32	26	24	1.5909	0.1694
33	24	30	-0.1312	0.0068
33	30	24	0.1312	-0.0025
34	27	28	-0.7111	-0.1245
34	28	27	0.7111	0.2285
35	27	29	-0.6081	-0.1486
35	29	27	0.6081	0.2229
36	30	31	-0.0437	0.0008
36	31	30	0.0437	0.0000
37	30	32	-0.0437	0.0008
37	32	30	0.0437	0.0000
38	30	33	-0.0437	0.0008
38	33	30	0.0437	0.0000
39	24	34	0.4631	-0.0241
39	34	24	-0.4618	0.0152
40	34	35	0.3489	0.1105
40	35	34	-0.3489	-0.0979
41	34	36	0.2111	-0.0223
41	36	34	-0.2111	0.0300
42	34	38	-0.4130	-0.4250
42	38	34	0.4130	0.4309
43	34	37	0.1148	-0.0284
43	37	34	-0.1148	0.0305
44	35	36	0.0389	-0.2421
44	36	35	-0.0389	0.2457
45	37	38	-0.2252	-0.0221
45	38	37	0.2252	0.0276
46	1	24	-0.3248	0.0657
46	24	1	0.3268	-0.0664
47	24	39	0.9249	-0.0251
47	39	24	-0.9186	0.0567
48	27	42	0.4065	-0.0883
48	42	27	-0.3983	0.1368
49	1	39	0.0851	0.0844
49	39	1	-0.0850	-0.0909
50	39	40	0.4145	0.0577
50	40	39	-0.4145	-0.0192
51	39	41	0.5891	-0.0234
51	41	39	-0.5891	0.0685
52	40	41	-0.2291	-0.2049
52	41	40	0.2291	0.2116
53	40	42	0.2235	0.0742
53	42	40	-0.2231	-0.0710
54	42	43	-0.0614	-0.0620
54	43	42	0.0614	0.0647
55	42	44	-0.0359	-0.0236
55	44	42	0.0359	0.0253
56	42	45	-0.0383	-0.0669
56	45	42	0.0383	0.0688
57	42	46	0.1471	-0.1232
57	46	42	-0.1471	0.1296

```
  58      46     47      -0.0303                  0.0003
  58      47     46       0.0303                  0.0003
  59      46     48      -0.0425                  0.0008
  59      48     46       0.0425                  0.0008

 >>TYPE CODE # FOR DESIRED ACTION
        1=PRINT BUS & GEN SOLUTIONS ONLY.
        2=PRINT LINE  POWERS, BUS & GEN SOLUTIONS,
             AND SYSTEM OPERATING COST.
        3=CREATE SOS FILE OF  SOLUTIONS
        4=CHANGE DATA OR TOLERANCES AND RERUN
        5=CHANGE TYPE PROBLEM OR SOLUTION METHOD,
             RERUN WITH EXISTING DATA
        6=NEW PROBLEM
        7=TERMINATE
 7
 STOP


.RUN EDSP50 WITH LIGHT SYSTEM LOAD OF 3.26 pu

               ****ECONOMIC DISPATCH EXECUTIVE PROGRAM****

>>ENTER TYPE PROBLEM
       0=LOAD FLOW PROBLEM
       1=ECONOMIC DISPATCH PROBLEM
0

>>ENTER TOTAL NUMBER OF (LINES, & BUSES).
59,48

>>WANT TO TYPE IN NEW(LINE DATA, BUS DATA, OR GENERATOR DATA,
    OR CALCULATE Y/ZBUS MATRIX)?
NNNNN

>>WANT A COPY OF (LDATA,BDATA,GDATA,OR SYSTEM CONVERGENCE CRITERIA)?
NYYN

**SYSTEM BUS DATA**
 BUS    TYPE      PG          QG          PD          QD         VMAG        VANG
  1      -2    1.10000     0.20000     0.00000     0.00000     1.05000     0.00000
  2       2    0.00000     0.00000     0.01000     0.00500     1.00000     0.00000
  3       1    0.00000     0.00000     0.35000     0.13000     1.00000     0.00000
  4       1    0.00000     0.00000     0.00000     0.00000     1.00000     0.00000
  5      -1    0.00000     0.00000     0.00000     0.00000     1.00000     0.00000
  6      -1    0.00000     0.00000     0.00000     0.00000     1.00000     0.00000
  7      -1    0.00000     0.00000     0.00000     0.00000     1.00000     0.00000
  8      -1    0.00000     0.00000     0.00000     0.00000     1.00000     0.00000
  9       1    0.00000     0.00000     0.00000     0.00000     1.00000     0.00000
 10       1    0.00000     0.00000     0.35000     0.21000     1.00000     0.00000
 11       2    0.00000     0.00000     0.07000     0.01500     1.00000     0.00000
 12       1    0.00000     0.00000     0.00000     0.00000     1.00000     0.00000
 13       1    0.00000     0.00000     0.40000     0.24000     1.00000     0.00000
 14       2    0.00000     0.00000     0.00000     0.00000     1.00000     0.00000
 15       2    0.00000     0.00000     0.37000     0.13000     1.00000     0.00000
 16      -2    1.65000     0.00000     0.00000     0.00000     1.01450     0.00000
 17       1    0.00000     0.00000     0.26000     0.04000     1.00000     0.00000
 18       1    0.00000     0.00000     0.00000     0.00000     1.00000     0.00000
 19       2    0.00000     0.00000     0.02000     0.01800     1.00000     0.00000
 20       1    0.00000     0.00000     0.00000     0.00000     1.00000     0.00000
 21       1    0.00000     0.00000     0.00000     0.00000     1.00000     0.00000
 22       2    0.00000     0.10000     0.10000     0.02000     1.00000     0.00000
 23       1    0.00000     0.00000     0.00000     0.00000     1.00000     0.00000
 24     1   0.00000   0.00000   0.00000   0.00000   1.00000   0.00000
 25    -2   0.00000   0.00000   0.00000   0.00000   1.00000   0.00000
 26    -3   0.70000   0.60000   0.00000   0.00000   1.00000   0.00000
 27     1   0.00000   0.00000   0.60000   0.30000   1.00000   0.00000
 28    -2   0.00000   0.00000   0.00000   0.00000   1.00000   0.00000
 29    -2   0.00000   0.00000   0.00000   0.00000   1.00000   0.00000
 30     1   0.00000   0.00000   0.00000   0.00000   1.00000   0.00000
 31    -1   0.00000   0.00000   0.00000   0.00000   1.00000   0.00000
 32    -1   0.00000   0.00000   0.00000   0.00000   1.00000   0.00000
 33    -1   0.00000   0.00000   0.00000   0.00000   1.00000   0.00000
 34     1   0.00000   0.00000   0.20000   0.03000   1.00000   0.00000
 35     1   0.00000   0.00000   0.05000   0.02000   1.00000   0.00000
 36     2   0.00000   0.00000   0.05000   0.01000   1.00000   0.00000
 37     2   0.00000   0.00000   0.22000   0.12000   1.00000   0.00000
 38     1   0.00000   0.00000   0.00000   0.00000   1.00000   0.00000
 39     1   0.00000   0.00000   0.00000   0.00000   1.00000   0.00000
 40     1   0.00000   0.00000   0.22000   0.01000   1.00000   0.00000
 41     2   0.00000   0.20000   0.15000   0.05000   1.00000   0.00000
 42     1   0.00000   0.00000   0.02000   0.01000   1.00000   0.00000
 43    -2   0.00000   0.00000   0.00000   0.00000   1.00000   0.00000
 44    -2   0.00000   0.00000   0.00000   0.00000   1.00000   0.00000
 45    -2   0.00000   0.00000   0.00000   0.00000   1.00000   0.00000
 46     2   0.00000   0.00000   0.02000   0.01000   1.00000   0.00000
 47    -2   0.00000   0.00000   0.00000   0.00000   1.00000   0.00000
 48    -2   0.00000   0.00000   0.00000   0.00000   1.00000   0.00000

**GENERATOR CONSTRAINTS**
BUS #    PGMAX      PGMIN        QGMAX      QGMIN
  1     1.1000     0.0000       0.2000    -2.0000
  2                             2.0000    -2.0000
  5     0.2510     0.0000
  6     0.2510     0.0000
  7     0.2510     0.0000
  8     0.2510     0.0000
 11                             0.8000    -2.0000
 14                             0.4000    -2.0000
 15                             0.4000    -2.0000
 16     1.6500     0.0000       2.0000    -2.0000
 19                             0.4500    -2.0000
 22                             0.2400    -2.0000
 25     1.0900     0.0000       2.0000    -2.0000
 26     1.5900     0.0000
 28     0.7700     0.0000       2.0000    -2.0000
 29     0.7900     0.0000       2.0000    -2.0000
 31     0.2490     0.0000
 32     0.2490     0.0000
 33     0.2490     0.0000
 36                             0.4000    -2.0000
 37                             0.4300    -2.0000
 41                             0.4000    -2.0000
 43     0.4810     0.0000       0.4500    -2.0000
 44     0.1480     0.0000       0.1500    -2.0000
 45     0.1320     0.0000       0.1400    -2.0000
 46                             0.3600    -2.0000
 47     0.2450     0.0000       0.2500    -2.0000
 48     0.4680     0.0000       0.4500    -2.0000

>>WANT TO CHANGE (LDATA,BDATA,GDATA, OR CONVERGENCE CRITERIA)?
NNNN

>>WANT A COPY OF (YBUS MATRIX, ZBUS MATRIX)?
NN
```

```
**SOLVING LOAD FLOW EQUATIONS BY  NR**
****NR CONVERGENCE IN  3 ITERATIONS
    ***OPTIMIZING COST FUNCTIONS***
          LAMBDA DETERMINED IN  20  ITERATIONS

**SOLVING LOAD FLOW EQUATIONS BY  NR**
****NR CONVERGENCE IN  2 ITERATIONS
    ***OPTIMIZING COST FUNCTIONS***
          LAMBDA DETERMINED IN  18  ITERATIONS

>>TYPE CODE # FOR DESIRED ACTION
       1=PRINT BUS & GEN SOLUTIONS ONLY.
       2=PRINT LINE  POWERS, BUS & GEN SOLUTIONS,
           AND SYSTEM OPERATING COST.
       3=CREATE SOS FILE OF  SOLUTIONS
       4=CHANGE DATA OR TOLERANCES AND RERUN
       5=CHANGE TYPE PROBLEM OR SOLUTION METHOD,
           RERUN WITH EXISTING DATA
       6=NEW PROBLEM
       7=TERMINATE
2

****SYSTEM BUS AND GENERATOR SOLUTIONS****
     ECON. DISPATCH CONVERGENCE IN  2 ITERATIONS
 ESP1=  0.0010, ESP2=  0.0010,  LAMBDA=  8.6383
SYSTEM LOSSES(PU)=  0.0456,  SYSTEM LOAD(PU)=  3.2600
BUS# BTYPE     IC      ITL       PG       QG       PD       QD      VMAG    VANG
  1   -2    0.0000  0.0220  1.1000 -0.0703  0.0000  0.0000  1.0000 -0.0379
  2    2    0.0000  0.0000  0.0000  0.0912  0.0100  0.0050  1.0000 -0.0586
  3    1    0.0000  0.0000  0.0000  0.0000  0.3500  0.1300  0.9947 -0.0647
  4    1    0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.9947 -0.0647
  5   -1    8.4448  0.0224  0.0000  0.0000  0.0000  0.0000  0.9947 -0.0647
  6   -1    8.4448  0.0224  0.0000  0.0000  0.0000  0.0000  0.9947 -0.0647
  7   -1    8.4448  0.0224  0.0000  0.0000  0.0000  0.0000  0.9947 -0.0647
  8   -1    8.4448  0.0224  0.0000  0.0000  0.0000  0.0000  0.9947 -0.0647
  9    1    0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.9966 -0.0702
 10    1    0.0000  0.0000  0.0000  0.0000  0.3500  0.2100  0.9907 -0.1271
 11    2    0.0000  0.0000  0.0000  0.2577  0.0700  0.0150  1.0000 -0.1148
 12    1    0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.9979 -0.0760
 13    1    0.0000  0.0000  0.0000  0.0000  0.2000  0.2000  0.9913 -0.0913
 14    2    0.0000  0.0000  0.0000  0.1524  0.0000  0.0000  1.0000 -0.0875
 15    2    0.0000  0.0000  0.0000  0.1597  0.3700  0.1300  1.0000 -0.1085
 16   -2    0.0000  0.0100  1.6500 -0.8335  0.0000  0.0000  1.0145  0.2412
 17    1    0.0000  0.0000  0.0000  0.0000  0.2600  0.0400  1.0376  0.1707
 18    1    0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  1.0134 -0.0283
 19    2    0.0000  0.0000  0.0000 -0.1606  0.0200  0.0180  1.0000 -0.0360
 20    1    0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  1.0095 -0.0394
 21    1    0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  1.0135 -0.0627
 22    2    0.0000  0.0000  0.0000  0.0472  0.1000  0.1000  1.0000 -0.0874
 23    1    0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  1.0057 -0.0767
 24    1    0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  1.0043 -0.0416
 25   -2    8.5614  0.0089  0.0000  0.0776  0.0000  0.1000  1.0000 -0.0416
 26   -3    8.5656  0.0084  0.3551  0.0705  0.0000  0.1000  1.0000  0.0000
 27    1    0.0000  0.0000  0.0000  0.0000  0.6000  0.3000  0.9871 -0.0910
 28   -2    8.5433  0.0110  0.0000  0.0690  0.0000  0.0000  1.0000 -0.0910
 29   -2    8.5471  0.0106  0.2005  0.0762  0.0000  0.0000  1.0000 -0.0550
 30    1    0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  1.0043 -0.0416
 31   -1    8.5617  0.0089  0.0000  0.0000  0.0000  0.0000  1.0043 -0.0416
 32   -1    8.5617  0.0089  0.0000  0.0000  0.0000  0.0000  1.0043 -0.0416
 33   -1    8.5617  0.0089  0.0000  0.0000  0.0000  0.0000  1.0043 -0.0416
 34    1    0.0000  0.0000  0.0000  0.0000  0.2000  0.0300  1.0071 -0.0393
 35    1    0.0000  0.0000  0.0000  0.0000  0.0500  0.0200  1.0020 -0.0450
 36   2     0.0000  0.0000  0.0000  0.0339  0.0500  0.1100  1.0000 -0.0457
 37   2     0.0000  0.0000  0.0000  0.0714  0.2200  0.2200  1.0000 -0.0486
 38   1     0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  1.0110 -0.0320
 39   1     0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  1.0004 -0.0504
 40   1     0.0000  0.0000  0.0000  0.0000  0.2200  0.0100  0.9986 -0.0881
 41   2     0.0000  0.0000  0.0000  0.0711  0.1500  0.0500  1.0000 -0.0816
 42   1     0.0000  0.0000  0.0000  0.0000  0.0200  0.0100  0.9975 -0.0906
 43  -2     8.5268  0.0129  0.0000  0.0076  0.0000  0.0000  1.0000 -0.0906
 44  -2     8.5268  0.0129  0.0000  0.0029  0.0000  0.0000  1.0000 -0.0906
 45  -2     8.5268  0.0129  0.0000  0.0081  0.0000  0.0000  1.0000 -0.0906
 46   2     0.0000  0.0000  0.0000  0.0251  0.0200  0.0100  1.0000 -0.0939
 47  -2     8.5265  0.0129  0.0000  0.0000  0.0000  0.0000  1.0000 -0.0939
 48  -2     8.5265  0.0129  0.0000  0.0000  0.0000  0.0000  1.0000 -0.0939

****SYSTEM OPERATING COST****
GEN.#  1=     0.00P.U.
GEN.#  5=     0.00P.U.
GEN.#  6=     0.00P.U.
GEN.#  7=     0.00P.U.
GEN.#  8=     0.00P.U.
GEN.# 16=     0.00P.U.
GEN.# 25=     0.00P.U.
GEN.# 26=     3.01P.U.
GEN.# 28=     0.00P.U.
GEN.# 29=     1.66P.U.
GEN.# 31=     0.00P.U.
GEN.# 32=     0.00P.U.
GEN.# 33=     0.00P.U.
GEN.# 43=     0.00P.U.
GEN.# 44=     0.00P.U.
GEN.# 45=     0.00P.U.
GEN.# 47=     0.00P.U.
GEN.# 48=     0.00P.U.
TOTAL SYSTEM OPERATING COST =       4.67P.U.

****LINE POWER FLOW****
LINE#    SB     EB     REAL POWER    REACTIVE POWER
  1       1      2       0.1083            0.0011
  1       2      1      -0.1083            0.0011
  2       1      3       0.2517            0.0530
  2       3      1      -0.2517           -0.0460
  3       2      3       0.0983            0.0851
  3       3      2      -0.0983           -0.0840
  4       3      4      -0.0000           -0.0000
  4       4      3       0.0000            0.0000
  5       4      5      -0.0000            0.0000
  5       5      4       0.0000            0.0000
  6       4      6      -0.0000            0.0000
  6       6      4       0.0000            0.0000
  7       4      7      -0.0000            0.0000
  7       7      4       0.0000            0.0000
  8       4      8      -0.0000            0.0000
  8       8      4       0.0000            0.0000
  9       1      9       0.5129           -0.0295
  9       9      1      -0.5100            0.0377
 10       9     10       0.0797            0.0107
 10      10      9      -0.0797           -0.0061
 11       9     11       0.3403           -0.0183
 11      11      9      -0.3403            0.0336
 12      10     11      -0.2703           -0.2039
 12      11     10       0.2703            0.2091
 13       9     12       0.0900           -0.0301
 13      12      9      -0.0900            0.0257
```

14	12	13	0.1402	0.0627
14	13	12	-0.1402	-0.0601
15	12	14	0.0598	-0.0104
15	14	12	-0.0598	0.0111
16	12	15	0.3700	-0.0176
16	15	12	-0.3700	0.0297
17	13	14	-0.0598	-0.1399
17	14	13	0.0598	0.1413
18	12	20	-0.4800	-0.0603
18	20	12	0.4832	0.0707
19	16	17	1.6500	-0.8335
19	17	16	-1.6379	0.5502
20	17	18	1.3779	-0.5902
20	18	17	-1.3526	-0.6270
21	18	20	0.6920	0.2441
21	20	18	-0.6920	-0.2354
22	18	19	0.0524	0.0913
22	19	18	-0.0524	-0.0897
23	19	20	0.0324	-0.0888
23	20	19	-0.0324	0.0898
24	20	21	0.1007	-0.1461
24	21	20	-0.1000	-0.0560
25	18	38	0.6082	0.2916
25	38	18	-0.6079	-0.3484
26	21	22	0.0477	0.0268
26	22	21	-0.0477	-0.0252
27	21	23	0.0523	0.0293
27	23	21	-0.0523	-0.0283
28	22	23	-0.0523	-0.0276
28	23	22	0.0523	0.0283
29	20	24	0.1406	0.2211
29	24	20	-0.1404	-0.2289
30	24	27	0.3917	0.1479
30	27	24	-0.3917	-0.1262
31	24	25	-0.0000	0.0225
31	25	24	0.0000	-0.0224
32	24	26	-0.3551	0.0444
32	26	24	0.3551	-0.0295
33	24	30	-0.0000	0.0000
33	30	24	0.0000	0.0000
34	27	28	-0.0000	-0.0681
34	28	27	0.0000	0.0690
35	27	29	-0.2005	-0.0681
35	29	27	0.2005	0.0762
36	30	31	-0.0000	0.0000
36	31	30	0.0000	0.0000
37	30	32	-0.0000	0.0000
37	32	30	0.0000	0.0000
38	30	33	-0.0000	0.0000
38	33	30	0.0000	0.0000
39	24	34	-0.0878	-0.0789
39	34	24	0.0879	0.0635
40	34	35	0.0616	0.0549
40	35	34	-0.0616	-0.0542
41	34	36	0.0384	0.0425
41	36	34	-0.0384	-0.0420
42	34	38	-0.4508	-0.2388
42	38	34	0.4508	0.2431
43	34	37	0.0629	0.0480
43	37	34	-0.0629	-0.0471
44	35	36	0.0116	0.0342
44	36	35	-0.0116	-0.0341
45	37	38	-0.1571	-0.1016
45	38	37	0.1571	0.1053

```
 46      1     24       0.0271               -0.0517
 46     24      1      -0.0271                0.0394
 47     24     39       0.2187                0.0536
 47     39     24      -0.2184               -0.0566
 48     27     42      -0.0078               -0.0377
 48     42     27       0.0079                0.0380
 49      1     39       0.2000               -0.0433
 49     39      1      -0.1996                0.0383
 50     39     40       0.1737                0.0114
 50     40     39      -0.1737               -0.0049
 51     39     41       0.2442                0.0069
 51     41     39      -0.2442                0.0007
 52     40     41      -0.0942               -0.0198
 52     41     40       0.0942                0.0204
 53     40     42       0.0479                0.0147
 53     42     40      -0.0479               -0.0145
 54     42     43      -0.0000               -0.0075
 54     43     42       0.0000                0.0076
 55     42     44      -0.0000               -0.0029
 55     44     42       0.0000                0.0029
 56     42     45      -0.0000               -0.0081
 56     45     42       0.0000                0.0081
 57     42     46       0.0200               -0.0150
 57     46     42      -0.0200                0.0151
 58     46     47      -0.0000                0.0000
 58     47     46       0.0000                0.0000
 59     46     48      -0.0000                0.0000
 59     48     46       0.0000                0.0000

>>TYPE CODE # FOR DESIRED ACTION
       1=PRINT BUS & GEN SOLUTIONS ONLY.
       2=PRINT LINE  POWERS, BUS & GEN SOLUTIONS,
            AND SYSTEM OPERATING COST.
       3=CREATE SOS FILE OF  SOLUTIONS
       4=CHANGE DATA OR TOLERANCES AND RERUN
       5=CHANGE TYPE PROBLEM OR SOLUTION METHOD,
            RERUN WITH EXISTING DATA
       6=NEW PROBLEM
       7=TERMINATE
7
STOP

END OF EXECUTION
CPU TIME: 3:54.54        ELAPSED TIME: 17:48.93
EXIT
```

APPENDIX B

COMPUTER CODE LISTINGS

Computer Code EDSP50

```
C ECONOMIC DISPATCH EXECUTIVE PROGRAM
C       THIS PROGRAM PROVIDES LOAD FLOW SOLUTIONS AND ECONOMIC
C       DISPATCH FOR POWER SYSTEMS WITH A MAXIMUM OF 50
C       BUSES AND 100 TRANSMISSION LINES. IT SOLVES FOR SYSTEM
C       LOAD,LOSSES,GENERATOR INCREMENTAL COSTS,REAL POWER
C       GENERATION,TRANSMISSION LINE REAL & REACTIVE POWER FLOWS.

        REAL LNGTH,LAMBDA,IC
        INTEGER ABORT,ANS,BDAT,BTYPE,BUSM,CE,CC,CG,CHC,CHB,CHG,CHL,
        1CL,CHY,CHZ,EB,GDAT,QDAT,MBUS,BGSOL,LPSOL,SB,ILDAT,TBDAT,
        1TGDAT,TBUSM
        DOUBLE PRECISION AJ1,AJ2,AJ3,AJ4,BJ1,BJ2,BJ3,BJ4,
        1CALP,CALQ,DQ,DVANG,DVMAG,G,H,ITL,P,C,VANG,VMAG,
        2XX,YANG,YMAG,YY,YYSHT,ZANG,ZMAG,YSER
        COMMON/ /       AJ1(50,50),     AJ2(50,50),     AJ3(50,50),
        1AJ4(50,50),    ALPHA(50),      BDAT,           BETA(50),
        1BGSOL,         BJ1(50,50),     BJ2(50,50),     BJ3(50,50),
        1BJ4(50,50),    BTYPE(50),      BUSM,           CALP(50),
        1CALQ(50),      CB,CC,            CG,             CHB,
        1CHG,           CHL,            CL,             CHY,
        1CHZ,           DQ(50),         DVANG(50),      DVMAG(50),
        1EB(100),        ESP1,           ESP2,           G(50),
        1GAMMA(50),     GDAT,           H(50),          IC(50),
        1II,            ITER,           ITL(50),        KINV,
        1LAMBDA,        LDAT,           LITER,
        1LNGTH(100),     LNO,            LOSS,           LPSOL,
        1LSET,          MBUS,           N5,             NSWGB,
        1P(50),         PD(50),         PG(50),         PGMAX(50),
        1PGMIN(50),     Q(50),          QD(50),QDAT,       QG(50),
        1QGMAX(50),     QGMIN(50),      SB(100),         YYSHT(100,2),
        1TOL,TLDAT,     TPD,TBDAT,      TPL,TGDAT,      VANG(50),
        1VMAG(50),      XX(50,50),      YANG(50,50),
        1YMAG(50,50),   YSHT(100,2),     YY(50,50),       ZANG(50,50),
        1ZMAG(50,50),   ZSER(100,2),     YSER(100,2),       ABORT,IBT(50)
        DATA
        1ABORT/0/,         BDAT/0/,           BUSM/0/,           CB/'N'/,
        1CG/'N'/,             CHB/'N'/,             CHC/'N'/,
        1CHG/'N'/,            CHL/'N'/,             CL/'N'/,
        1CHY/'N'/,            CHZ/'N'/,             ESP1/0.001/,
        1ESP2/0.001/,         GDAT/0/,            ITER/50/,            KINV/0/,
        1LDAT/0/,          LITER/5/,          LOSS/'N'/,
        1TOL/0.0001/,       QDAT/0/

        CALL ERRSET(0)
10      TYPE 1000
1000    FORMAT(15X'****ECONOMIC DISPATCH EXECUTIVE PROGRAM****')

C SYSTEM INITIALIZATION
20      TYPE 1010
1010    FORMAT(/1X,'>>ENTER TYPE PROBLEM'/8X,'0=LOAD FLOW PROBLEM'/
        18X,'1=ECONOMIC DISPATCH PROBLEM')
22      ACCEPT 2000,KINV
2000    FORMAT(I1)
        IF((KINV.EQ.0).OR.(KINV.EQ.1)) GO TO 24
        TYPE 1012
1012    FORMAT(1X,'WRONG ENTRY, MUST BE 0 OR 1, TRY AGAIN')
        GO TO 22
24      IF(KINV)30,30,26
26      TYPE 1014
1014    FORMAT(/1X,'>>IS SYSTEM LOSSLESS?')
28      ACCEPT 2001,LOSS
```

```
2001     FORMAT(1A1)
         IF((LOSS.EQ.'N').OR.(LOSS.EQ.'Y'))GO TO 30
         TYPE 1013
1013     FORMAT(1X,'WRONG ENTRY, MUST BE "Y" OR "N", TRY AGAIN.')
         GO TO 28
30       IF((LDAT.EQ.1).AND.(BDAT.EQ.1).AND.(GDAT.EQ.1))
         1GO TO 70
         TYPE 1020
1020     FORMAT(/1X,'>>ENTER TOTAL NUMBER OF (LINES, & BUSES).')
36       ACCEPT 2004,LNO,NBUS
2004     FORMAT(2I)
         IF(((LNO.GE.1).AND.(LNO.LE.100)).AND.((NBUS.GE.1)
         1.AND.(NBUS.LE.50)))GO TO 40
         TYPE1016
1016     FORMAT(1X,'WRONG ENTRY,TOTAL LINES=1 TO 100 &TOTAL BUSES=1 TO 50,
         1TRY AGAIN.')
         GO TO 36

C CHECK METHOD OF DATA INPUT(SOS FILE/CRT/TTY)
40       TYPE 1030
1030     FORMAT(/1X,'>>WANT TO TYPE IN NEW(LINE DATA, BUS DATA, OR
         1 GENERATOR DATA,'/5X,'OR CALCULATE Y/ZBUS MATRIX)?')
50       ACCEPT 2008,TLDAT,TBDAT,TGDAT,TBUSM
         IF(((TLDAT.EQ.'N').OR.(TLDAT.EQ.'Y')).AND.((TBDAT.EQ.'N')
         1.OR.(TBDAT.EQ.'Y')).AND.((TGDAT.EQ.'N').OR.(TGDAT.EQ.'Y')
         1.AND.(TBUSM.EQ.'N').OR.(TBUSM.EQ.'Y')))GO TO 60
         TYPE 1015
1015     FORMAT(1X,'WRONG ENTRY, MUST BE "N" OR "Y" FOR
         1EACH DATA GROUP,'/1x,'i.e. "NYN" means type in bus
         1data only,please use CAP N & Y,'/1x,'try again.')
2008     FORMAT(4A1)
         GO TO 50

C READ SYSTEM DATA FROM SOS FILE,TTY,OR CRT
60       IF(LDAT.EQ.0)CALL LDATA
         IF(ABORT.EQ.1)GO TO 320
         IF(TBUSM.EQ.'Y')GO TO 80
         OPEN(UNIT=24,DEVICE='DSK',ACCESS='SEQIN',DISPOSE='SAVE',
         1FILE='FOR24.DAT')
         DO 52 I=1,LNO
52       READ(24,4000)(YYSHT(I,1),YYSHT(I,2),YSER(I,1),YSER(I,2))
4000     FORMAT(4D)
         DO 54 I=1,NBUS
54       READ(24,4005)(YMAG(I,J),YANG(I,J),J=1,NBUS)
4005     FORMAT(2D)
         BUSM=1
         IF(KINV.EQ.0)GO TO 80
         DO 55 I=1,NBUS
55       READ(24,4005)(ZMAG(I,J),ZANG(I,J),J=1,NBUS)
         BUSM=2
80       IF(BDAT.EQ.0)CALL BDATA
         IF(ABORT.EQ.1)GO TO 320
90       IF(GDAT.EQ.0)CALL GDATA
         IF(ABORT.EQ.1)GO TO 320

C CHECK FOR PRINTOUT OF SYSTEM DATA
70       TYPE 1050
1050     FORMAT(/1X,'>>WANT A COPY OF (LDATA,BDATA,GDATA,
         1OR SYSTEM CONVERGENCE CRITERIA)?')
```

```
100       READ(5,2020)CL,CB,CG,CC
2020      FORMAT(4A1)
          IF(((CL.EQ.'N').OR.(CL.EQ.'Y')).AND.
          1((CB.EQ.'N').OR.(CB.EQ.'Y')).AND.
          2((CG.EQ.'N').OR.(CG.EQ.'Y')).AND.
          3((CC.EQ.'N').OR.(CC.EQ.'Y')))GO TO 110
          TYPE 1019
1019      FORMAT(1X,'WRONG ENTRY, MUST BE "N" OR "Y" FOR
          1EACH DATA GROUP.'/1X,'I.E."NYNY" MEANS TYPE OUT
          2BUS DATA AND CONVERGENCE CRITERIA ONLY.'/1X, 'TRY AGAIN.')
          GO TO 100
110       IF(CL.EQ.'Y')CALL LDATA
120       IF(CB.EQ.'Y')CALL BDATA
130       IF(CG.EQ.'Y')CALL GDATA

C         PRINT SYSTEM CONVERGENCE CRITERIA
140       IF(CC.EQ.'N')GO TO 145
142       WRITE(5,1065)LITER,TOL,ITER,ESP1,ESP2
1065      FORMAT(/1X,'**SYSTEM CONVERGENCE CRITERIA**'/10X,
          1'LOAD FLOW ITERATIONS=',I3/
          110X,'LOAD FLOW TOLERANCE=',E13.5/
          210X,'ECON DISP ITERATIONS=',I3/10X,'ESP1=',E13.5/10X,'ESP2=',
          3E13.5)

C CHECK FOR CHANGES TO CONVERGENCE CRITERIA OR SYSTEM DATA
145       TYPE 1070
1070      FORMAT(/1X,'>>WANT TO CHANGE
          1(LDATA,BDATA,GDATA, OR CONVERGENCE CRITERIA)?')
150       ACCEPT 2040,CHL,CHB,CHG,CHC
2040      FORMAT(4A1)
          IF(((CHC.EQ.'N').OR.(CHC.EQ.'Y')).AND.
          1((CHL.EQ.'N').OR.(CHL.EQ.'Y')).AND.
          1((CHB.EQ.'N').OR.(CHB.EQ.'Y')).AND.
          1((CHG.EQ.'N').OR.(CHG.EQ.'Y')))GO TO 160
          TYPE 1021
1021      FORMAT(1X,'WRONG ENTRY, MUST BE "N" OR "Y" FOR EACH DATA
          1GROUP, TRY AGAIN.')
          GO TO 150
160       IF(CHL.EQ.'Y')CALL LDATA
          IF(CHB.EQ.'Y')CALL BDATA
          IF(CHG.EQ.'Y')CALL GDATA
          IF(CHC.EQ.'N')GO TO 190
          TYPE 1090
1090      FORMAT(/1X,'**CONVERGENCE CRITERIA CHANGES**'/
          11x,'>>ENTER(VARIABLE #,NEW VALUE THEN "CR")'/
          110X,'1=LOAD FLOW ITERATIONS'/10X,'2=LOAD FLOW TOLERANCE'/
          210X,'3=ECON DISP ITERATIONS'/10X,'4=ESP1'/10X,'5=ESP2'/
          310X,'>>ENTER 0 THEN "CR" WHEN CHANGES CCMPLETED.')
170       READ(5,2050)NAME,X
2050      FORMAT(I,F)
          IF((NAME.GE.0).AND.(NAME.LE.5))GO TO 180
          TYPE 1100
1100      FORMAT(1X,'WRONG ENTRY,VARIABLE MUST BE 1-5, TRY AGAIN.')
          GO TO 170
180       IF(NAME.EQ.0)GO TO 190
          GO TO(181,182,183,184,185)NAME
181       LITER=IFIX(X)
          TYPE 1110,LITER
1110      FORMAT(8X,'LOAD FLOW ITERATIONS=',I3)
          GO TO 170
182       TOL=X
```

```
        TYPE 1120,TOL
1120    FORMAT(8X,'LOAD FLOW TOLERANCE=',E13.5)
        GO TO 170
183     ITER=IFIX(X)
        TYPE 1130,ITER
1130    FORMAT(8X,'OPT DISP ITERATIONS=',I3)
        GO TO 170
184     ESP1=X
        TYPE 1140,ESP1
1140    FORMAT(8X,'ESP1=',E13.5)
        GO TO 170
185     ESP2=X
        TYPE 1150,ESP2
1150    FORMAT(8X,'ESP2=',E13.5)
        GO TO 170

C CHECK FOR COPY OF Y/ZBUS MATRIX

C CHECK FOR COPY OF Y/ZBUS MATRIX
190     IF((BUSM.EQ.0).OR.((BUSM.NE.2).AND.(KINV.EQ.1)))CALL YBUSM
222     TYPE 1152
1152    FORMAT(/1X,'>>WANT A COPY OF (YBUS MATRIX, ZBUS MATRIX)?')
223     ACCEPT 2060,CHY,CHZ
        IF(KINV.EQ.0)CHZ='N'
2060    FORMAT(2A1)
        IF(((CHY.EQ.'N').OR.(CHY.EQ.'Y')).AND.((CHZ.EQ.'N').OR.
        1(CHZ.EQ.'Y')))GO TO 224
        TYPE 1021
        GO TO 223
224     IF ((CHY.EQ.'Y').OR.(CHZ.EQ.'Y'))CALL YBUSM
225     CONTINUE

C SYSTEM INITIALIZED STARTING LOAD FLOW CALCULATIONS
        LSET=0

C CALCULATING LOAD FLOW BY NEWTON-RAPHSON
        CALL NRLF
        IF(ABORT.EQ.1)GO TO 320
        IF (KINV.EQ.0)GO TO 320

C CALCULATING OPT DISP POWER BALANCE EQUATIONS
        CALL OPT
        GO TO 330

320     ABORT=0
330     TYPE1210
1210    FORMAT(/1X,'>>TYPE CODE # FOR DESIRED ACTION'/
        18X,'1=PRINT BUS & GEN SOLUTICNS ONLY.'/8X,'2=PRINT LINE
        2POWERS, BUS & GEN SOLUTIONS,'/13X,'AND SYSTEM
        2 OPERATING COST.'/8X,'3=CREATE SOS FILE OF
        2 SOLUTIONS'/8X,'4=CHANGE DATA OR TOLERANCES AND RERUN'/
        18X,'5=CHANGE TYPE PROBLEM OR SOLUTION METHOD,'/13X,'RERUN WITH
        1EXISTING DATA'/8X,'6=NEW PROBLEM'/8X,'7=TERMINATE')

340     ACCEPT 2000,ANS
        IF((ANS.GE.1).AND.(ANS.LE.7))GO TO 350
        TYPE 1220
1220    FORMAT(1X,'WRONG ENTRY, MUST BE 1 TO 7, TRY AGAIN.')
        GO TO 340
350     GO TO(361,362,380,363,365,367,500)ANS
```

```
361     N5=5
        BGSOL=1
        LPSOL=0
        CALL SOL
        GO TO 330
362     N5=5
        BGSOL=1
        LPSOL=1
        CALL SOL
        GO TO 330
363     LSET=0
        TBDAT='N'
        DO 65 I=1,NBUS
        IF(IBT(I).NE.2)GO TO 65
        BTYPE(I)=ISIGN(2,BTYPE(I))
        VMAG(I)=1.0
        IBT(I)=0
65      CONTINUE
        GO TO 145
        LSET=0
365     TBDAT='N'
        DO 66 I=1,NBUS
        IF(IBT(I).NE.2)GO TO 66
        BTYPE(I)=ISIGN(2,BTYPE(I))
        VMAG(I)=1.0
        IBT(I)=0
66      CONTINUE
        GO TO 10
367     LDAT=0
        BDAT=0
        GDAT=0
        QDAT=0
        LSET=0
        GO TO 10

C CREATING SOS FILE OF BUS & GENERATOR SOLUTIONS, AND
C LINE POWER FLOW.

380     OPEN(UNIT=1,DEVICE='DSK',ACCESS='SEQOUT',
        1DISPOSE='SAVE',FILE='SOL.DAT')
        N5=1
        BGSOL=1
        LPSOL=1
        CALL SOL
        CLOSE(UNIT=1)
        N5=5
        GO TO 330

500     CONTINUE
        STOP
        END


        SUBROUTINE LDATA
C SUBROUTINE TO READ, WRITE, AND CHANGE LINE DATA

        REAL LNGTH
        INTEGER ABORT,BUSN,CL,CNL,EB,SB,TLDAT
        DOUBLE PRECISION AJ1,AJ2,AJ3,AJ4,BJ1,BJ2,BJ3,BJ4,
        1CALP,CALQ,DQ,DVANG,DVMAG,G,H,ITL,P,Q,VANG,VMAG,
         2XX,YANG,YMAG,YY,YYSHT,ZANG,ZMAG,YSER
         COMMON/ /        AJ1(50,50),    AJ2(50,50),    AJ3(50,50),
         1AJ4(50,50),     ALPHA(50),     BDAT,          BETA(50),
         1BGSOL,          BJ1(50,50),    BJ2(50,50),    BJ3(50,50),
         1BJ4(50,50),     BTYPE(50),     BUSM,          CALP(50),
         1CALQ(50),       CB,CC,            CG,            CMB,
         1CHG,            CHL,           CL,            CMY,
         1CMZ,            DQ(50),        DVANG(50),     DVMAG(50),
         1EB(100),         ESP1,          ESP2,          G(50),
         1GAMMA(50),      GDAT,          H(50),         IC(50),
         1II,             ITER,          ITL(50),       KINV,
         1LAMBDA,         LDAT,          LITER,
         1LNGTH(100),      LNO,           LOSS,          LPSOL,
         1LSET,           MBUS,          N5,            NSNGB,
         1P(50),          PD(50),        PG(50),        PGMAX(50),
         1PGMIN(50),      Q(50),         QD(50),QDAT,       QG(50),
         1QGMAX(50),      QGMIN(50),     SB(100),        YYSHT(100,2),
         1TOL,TLDAT,      TPD,TBDAT,     TPL,TGDAT,     VANG(50),
         1VMAG(50),       XX(50,50),     YANG(50,50),
         1YMAG(50,50),    YSHT(100,2),    YY(50,50),     ZANG(50,50),
         1ZMAG(50,50),    ZSER(100,2),    YSER(100,2),   ABORT

         ILD=0
         IF(LDAT.EQ.1)GO TO 50
         IF(TLDAT.EQ.'Y')GO TO 20

C READ LINE DATA FOR SOS FILE FOR20.DAT
         OPEN(UNIT=20,DEVICE='DSK',ACCESS='SEQIN',DISPOSE='SAVE',
         1FILE='FOR20.DAT')
10       DO 5 I=1,LNO
5        READ(20,*)K,SB(I),EB(I),LNGTH(I),YSHT(I,1),YSHT(I,2),
         1ZSER(I,1),ZSER(I,2)
         CLOSE(UNIT=20)
2000     FORMAT(3I,5F)
         IF(K.EQ.LNO)GO TO 40
         TYPE 8
8        FORMAT(1X,'TOTAL NUMBER OF LINES ENTERED DOES NOT EQUAL
         1NUMBER OF DATA LINES IN FOR20.DAT')
         ABORT=1
         LDAT=0
         RETURN

C READ LINE DATA FROM CRT/TTY
20       OPEN(UNIT=20,DEVICE='DSK',ACCESS='SEQOUT',DISPOSE='SAVE',
         1FILE='FOR20.DAT')
         IF(ILD.EQ.1)GO TO 21
         TYPE 1000
1000     FORMAT(/1X,'**SYSTEM LINE DATA**'/1X,'>>ENTER (LINE #,
         1SB #,EB #, LINE LENGTH, YSHUNT G,YSHUNT B,
         2ZSERIES R,ZSERIES X)')
21       DO 30 I=1,LNO
         IF(ILD.EQ.1)GO TO 25
22       ACCEPT 2002,K,SB(I),EB(I),LNGTH(I),YSHT(I,1),YSHT(I,2),
         1ZSER(I,1),ZSER(I,2)
2002     FORMAT(3I,5F)
         IF(K.EQ.I)GO TO 25
         TYPE 1002,I
1002     FORMAT(1X,'WRONG ENTRY, LINE # SHOULD=',I3,', TRY AGAIN')
         GO TO 22
25       WRITE(20,1004)I,SB(I),EB(I),LNGTH(I),YSHT(I,1),YSHT(I,2),
         1ZSER(I,1),ZSER(I,2)
```

```
1004      FORMAT(3I3,5E12.5)
30        CONTINUE
          CLOSE(UNIT=20)
          IF(ILD.EQ.1)GO TO 80
40        BUSM=0
          LDAT=1

C CHECK FOR LINE DATA PRINT OUT
50        IF(CL.EQ.'N')GO TO 70
          TYPE 1010
1010      FORMAT(/1X,'**SYSTEM LINE DATA**'/1X,'LINE #',1X,
          1'SB',1X,'EB',4X,'LENGTH',12X,'Y SHUNT',19X,'Z SERIES')
          DO 60 I=1,LNO
60        TYPE 1020,I,SB(I),EB(I),LNGTH(I),YSHT(I,1),YSHT(I,2),
          1ZSER(I,1),ZSER(I,2)
1020      FORMAT(1X,I3,2X,2I3,1X,E12.5,2(1X,2E12.5))
          CL='N'

          IF(ILD.EQ.1)GO TO 80
C CHECK FOR LINE DATA CHANGES
70        IF(CHL.EQ.'N')GO TO 180
          TYPE 1030
1030      FORMAT(/1X,'** LINE DATA CHANGES**'/
          11x,'>>ENTER (VARIABLE #, LINE #, NEW VALUE THEN "CR")'/
          110X,'1=STARTING BUS #'/10X,'2=ENDING BUS #'/10X,'3=LINE
          1LENGTH'/10X,'4=Y SHUNT G'/10X,'5=Y SHUNT B'/10X,'6=Z SERIES R'/
          210X,'7=Z SERIES X'/10X,'8=RETYPE COPY OF ALL LINE DATA'/10X,'
          19=CHANGE TOTAL NUMBER OF LINES'/10X,'10=UPDATE SOS
          1FILE WITH CHANGES'/10X,'>>ENTER 0 THEN
          1"CR" WHEN CHANGES COMPLETED')
80        READ(5,2010)NAME,I,X
2010      FORMAT(2I,F)
          IF(NAME.EQ.0)GO TO 179
          IF(NAME.EQ.8)GO TO 172
          IF(NAME.EQ.9)GO TO 173
          IF(NAME.EQ.10)GO TO 175
          IF(((NAME.GE.1).AND.(NAME.LE.7)).AND.((I.GE.0).AND.(I.LE.LNO)))
          1GO TO 100
          TYPE 1040,LNO
1040      FORMAT(1X,'WRONG ENTRY, VARIABLE # MUST BE 1-7,'/
          11x,'LINE # MUST BE 1 TO',I3,', TRY AGAIN')
          GO TO 80
100       BUSM=0
          GO TO(110,120,130,140,150,160,170)NAME
110       SB(I)=IFIX(X)
          TYPE 1060, I,SB(I)
1060      FORMAT(1X,'SB(LINE #',I3,')=',I3)
          GO TO 80
120       EB(I)=IFIX(X)
          TYPE 1070,I,EB(I)
1070      FORMAT(1X,'EB(LINE #',I3,')=',I3)
          GO TO 80
130       LNGTH(I)=X
          TYPE 1080,I,LNGTH(I)
1080      FORMAT(1X,'LENGTH(LINE #',I3,')=',E12.5)
          GO TO 80
140       YSHT(I,1)=X
          TYPE 1090,I,YSHT(I,1)
1090      FORMAT(1X,'YSHT G (LINE #',I3,')=',E12.5)
          GO TO 80
150       YSHT(I,2)=X
```

```
        TYPE 1100,I,YSHT(I,2)
1100    FORMAT(1X,'YSHT B (LINE #',I3,')=',E12.5)
        GO TO 80
160     ZSER(I,1)=X
        TYPE 1110,I,ZSER(I,1)
1110    FORMAT(1X,'ZSER R (LINE #',I3,')=',E12.5)
        GO TO 80
170     ZSER(I,2)=X
        TYPE 1120,I,ZSER(I,2)
1120    FORMAT(1X,'ZSER X (LINE #',I3,')=',E12.5)
        GO TO 80
172     ILD=1
        CL='Y'
        GO TO 50
173     LNO=I
        BUSM=0
        TYPE 1130,I
1130    FORMAT(1X,'TOTAL NUMBER OF LINES =',I3)
        GO TO 80
175     ILD=1
        GO TO 20
179     ILD=0
        CHL='N'
180     RETURN
        END


        SUBROUTINE BDATA
C SUBROUTINE TO READ,WRITE,AND CHANGE BUS DATA
        INTEGER ABORT,BDAT,BTYPE,CB,CHB,MBUS,TBDAT
        DOUBLE PRECISION AJ1,AJ2,AJ3,AJ4,BJ1,BJ2,BJ3,BJ4,
        1CALP,CALQ,DQ,DVANG,DVMAG,G,H,ITL,P,Q,VANG,VMAG,
        2XX,YANG,YMAG,YY,YYSHT,ZANG,ZMAG,YSER
        COMMON/ /        AJ1(50,50),    AJ2(50,50),    AJ3(50,50),
        1AJ4(50,50),     ALPHA(50),     BDAT,          BETA(50),
        1BGSOL,          BJ1(50,50),    BJ2(50,50),    BJ3(50,50),
        1BJ4(50,50),     BTYPE(50),     BUSM,          CALP(50),
        1CALQ(50),       CB,CC,           CG,            CHB,
        1CHG,            CHL,           CL,            CMY,
        1CMZ,            DQ(50),        DVANG(50),     DVMAG(50),
        1EB(100),         ESP1,          ESP2,          G(50),
        1GAMMA(50),      GDAT,          H(50),         IC(50),
        1II,             ITER,          ITL(50),       KINV,
        1LAMBDA,         LDAT,          LITER,
        1LNGTH(100),      LNO,           LOSS,          LPSOL,
        1LSET,           MBUS,          NS,            NSWGB,
        1P(50),          PD(50),        PG(50),        PGMAX(50),
        1PGMIN(50),      Q(50),         QD(50),QDAT,       QG(50),
        1QGMAX(50),      QGMIN(50),     SB(100),        YYSHT(100,2),
        1TOL,TLDAT,      TPD,TBDAT,     TPL,TGDAT,     VANG(50),
        1VMAG(50),       XX(50,50),     YANG(50,50),
        1YMAG(50,50),    YSHT(100,2),    YY(50,50),     ZANG(50,50),
        1ZMAG(50,50),    ZSER(100,2),    YSER(100,2),   ABORT

        IBD=0
        IF(BDAT.EQ.1)GO TO 50
        IF(TBDAT.EQ.'Y')GO TO 20

C READ BUS DATA FROM SOS FILE FOR21.DAT
```

```
5       OPEN(UNIT=21,DEVICE='DSK',ACCESS='SEQIN',DISPOSE='SAVE',
        1FILE='FOR21.DAT')
10      DO 15 I=1,MBUS
15      READ(21,*)K,BTYPE(I),PG(I),QG(I),PD(I),QD(I),
        1VMAG(I),VANG(I)
2000    FORMAT(2I,6F)
        CLOSE(UNIT=21)
        IF(K.EQ.MBUS)GO TO 45
        TYPE 500
500     FORMAT(1X,'TOTAL NUMBER OF BUSES ENTERED DOES NOT
        1EQUAL NUMBER OF DATA LINES IN FOR21.DAT')
        BDAT=0
        ABORT=1
        RETURN



C READ BUS DATA FROM CRT/TTY
19      OPEN(UNIT=21,DEVICE='DSK',ACCESS='SEQOUT',DISPOSE='SAVE',
        1FILE='FOR21.DAT')
        IF(IBD.EQ.1)GO TO 21
20      TYPE1000
1000    FORMAT(/1X,'**SYSTEM BUS DATA**'/1X,'>>ENTER(BUS#,
        1 BUS TYPE,PG,QG,PD,QD,VMAG,VANG)')
21      DO 30 I=1,MBUS
        IF(IBD.EQ.1)GO TO 25
22      ACCEPT 2000,K,BTYPE(I),PG(I),QG(I),PD(I),QD(I),
        1VMAG(I),VANG(I)
        IF(K.EQ.I)GO TO 25
        TYPE 1005,I
1005    FORMAT(1X,'WRONG ENTERY, BUS # SHOULD=',I3,',TRY AGAIN.')
        GO TO 22
25      WRITE(21,1010)I,BTYPE(I),PG(I),QG(I),PD(I),QD(I),
        1VMAG(I),VANG(I)
1010    FORMAT(2I3,6F10.5)
30      CONTINUE
        CLOSE(UNIT=21)
45      BDAT=1
        IF(IBD.EQ.1)GO TO 80

C CHECK FOR PRINTOUT OF BUS DATA
50      IF(CB.EQ.'N')GO TO 70
        TYPE 1020
1020    FORMAT(/1X,'**SYSTEM BUS DATA**'/2X,'BUS',3X,'TYPE',
        15X,'PG',8X,'QG',8X,'PD',8X,'QD',7X,'VMAG',6X,'VANG')
        DO 60 I=1,MBUS
        TYPE 1030,I,BTYPE(I),PG(I),QG(I),PD(I),QD(I),
        1VMAG(I),VANG(I)
1030    FORMAT(1X,I3,4X,I3,6F10.5)
60      CONTINUE
        CB='N'
        IF(IBD.EQ.1)GO TO 80

C CHECK FOR BUS DATA CHANGES
70      IF(CHB.EQ.'N')GO TO 170
        TYPE 1040
1040    FORMAT(/1X,'**BUS DATA CHANGES**'/
        11X,'>>ENTER(VARIABLE #, BUS #, NEW VALUE THEN "CR"'/
        110X,'1=BUS TYPE'/10X,'2=PG'/10X,'3=QG'/10X,'4=PD'
        1/10X,'5=QD'/10X,'6=VMAG'/10X,'7=VANG'
        1/10X,'8=RETYPE COPY OF ALL BUS DATA'
        1/10X,'9=CHANGE TOTAL NUMBER OF BUSES'/10X,
        1' 10=REENTER ORIGINAL BUS DATA FROM SOS FILE'/10X,
        2'11=UPDATE SOS FILE WITH CHANGES'/10X,
        1'>>ENTER 0 THEN "CR" WHEN CHANGES COMPLETED.')
80      READ(5,2010)NAME,I,X
2010    FORMAT(2I,F)
        IF(NAME.EQ.0)GO TO 170
        IF(NAME.EQ.9)GO TO 164
        IF(NAME.EQ.10)GO TO 165
        IF(NAME.EQ.11)GO TO 166
        IF(((NAME.GE.1).AND.(NAME.LE.8)).AND.
        1((I.GE.0).AND.(I.LE.MBUS)))GO TO 100
        TYPE 1050,MBUS
1050    FORMAT(1X,'WRONG ENTRY, VARIABLE NAME MUST BE 1-8,'/
        11X,'BUS # MUST BE 1 TO',I3,',TRY AGAIN.')
        GO TO 80
100     GO TO(110,120,130,140,150,160,162,163)NAME
110     BTYPE(I)=IFIX(X)
        TYPE 1060,I,BTYPE(I)
1060    FORMAT(1X,'BUS TYPE(BUS #',I3,')=',I3)
        GO TO 80
120     PG(I)=X
        TYPE 1070,I,PG(I)
1070    FORMAT(1X,'PG(BUS #',I3,')=',E12.5)
        GO TO 80
130        QG(I)=X
        TYPE 1080,I,QG(I)
1080    FORMAT(1X,'QG(BUS #',I3,')=',E12.5)
        GO TO 80
140     PD(I)=X
        TYPE 1090,I,PD(I)
1090    FORMAT(1X,'PD(BUS #',I3,')=',E12.5)
        GO TO 80
150     QD(I)=X
        TYPE 1100,I,QD(I)
1100    FORMAT(1X,'QD(BUS #',I3,')=',E12.5)
        GO TO 80
160     VMAG(I)=X
        TYPE 1110,I,VMAG(I)
1110    FORMAT(1X,'VMAG(BUS #',I3,')=',E12.5)
        GO TO 80
162     VANG(I)=X
        TYPE 1111,I,VANG(I)
1111    FORMAT(1X,'VANG(BUS#',I3,'=',E12.5)
        GO TO 80
163     CB='Y'
        IBD=1
        GO TO 50
164     MBUS=I
        TYPE 1120,I
1120    FORMAT(1X,'TOTAL NUMBER OF BUSES =',I3)
        GO TO 80
165     IBD=1
        LSET=0
        GO TO 5
166     IBD=1
        GO TO 19

170     IBD=0
        CHB='N'
```

```
      RETURN
      END


      SUBROUTINE GDATA
C SUBROUTINE TO READ,WRITE,AND CHANGE GENERATOR DATA

      INTEGER BTYPE,CG,CHG,GDAT,GNO,MBUS,TGDAT,QDAT
      DOUBLE PPECISION AJ1,AJ2,AJ3,AJ4,BJ1,BJ2,BJ3,BJ4,
     1CALP,CALQ,DC,DVANG,DVMAG,G,H,ITL,P,C,VANG,VMAG,
     2XX,YANG,YMAG,YY,YYSHT,ZANG,ZMAG,YSER
      COMMON/ /       AJ1(50,50),   AJ2(50,50),   AJ3(50,50),
     1AJ4(50,50),     ALPHA(50),    BDAT,         BETA(50),
     1BGSOL,          BJ1(50,50),   BJ2(50,50),   BJ3(50,50),
     1BJ4(50,50),     BTYPE(50),    BUSM,         CALP(50),
     1CALQ(50),       CB,CC,          CG,           CHB,
     1CHG,            CHL,          CL,           CHY,
     1CHZ,            DQ(50),       DVANG(50),    DVMAG(50),
     1EB(100),         ESP1,         ESP2,         G(50),
     1GAMMA(50),      GDAT,         H(50),        IC(50),
     1II,             ITER,         ITL(50),      KINV,
     1LAMBDA,         LDAT,         LITER,
     1LNGTH(100),      LNO,          LOSS,         LPSOL,
     1LSET,           MBUS,         N5,           NSWGB,
     1P(50),          PD(50),       PG(50),       PGMAX(50),
     1PGMIN(50),      Q(50),        QD(50),QDAT,      QG(50),
     1QGMAX(50),      QGMIN(50),    SB(100),       YYSHT(100,2),
     1TOL,TLDAT,      TPD,TBDAT,    TPL,TGDAT,    VANG(50),
     1VMAG(50),       XX(50,50),    YANG(50,50),
     1YMAG(50,50),    YSHT(100,2),   YY(50,50),    ZANG(50,50),
     1ZMAG(50,50),    ZSER(100,2),   YSER(100,2),   ABORT

      IGB=0
      IF(GDAT.EQ.1)GO TO 56
      IF(KINV.EQ.0)GO TO 25
      IF(TGDAT.EQ.'Y')GO TO 20

C READ GENERATOR DATA FROM SOS FILE FOR22.DAT
      OPEN(UNIT=22,DEVICE='DSK',ACCESS='SEQIN',DISPOSE='SAVE',
     1FILE='FOR22.DAT')
      GNO=0
10    DO 15 I=1,MBUS
      IF((BTYPE(I)).GE.0)GO TO 15
      READ(22,*)K,ALPHA(I),BETA(I),GAMMA(I)
2000  FORMAT(I,3F)
      GNO=GNO+1
15    CONTINUE
      CLOSE(UNIT=22)
      GO TO 40

C READ GENERATOR DATA FROM CRT/TTY
20    OPEN(UNIT=22,DEVICE='DSK',ACCESS='SEQOUT',DISPOSE='SAVE',
     1FILE='FOR22.DAT')
      IF(IGB.EQ.1)GO TO 29
      TYPE 1000
1000  FORMAT(/1X,'**GENERATCR DATA**'/1X,'>>ENTER(BUS#,ALPHA,
     1BETA,GAMMA)')
29    DO 30 I=1,MBUS
      IF(BTYPE(I).GE.0)GO TO 30
      IF(IGB.EQ.1)GO TO 33
```

```
32      ACCEPT 2002,K,ALPHA(I),BETA(I),GAMMA(I)
2002    FORMAT(I,3F)
        IF(K.EQ.I)GO TO 31
        TYPE 1010,I
1010    FORMAT(1X,'URONG ENTRY, BUS # SHOULD=',I3,', TRY AGAIN.')
        GO TO 32
31      GNO=GNO+1
33      WRITE(22,2003)I,ALPHA(I),BETA(I),GAMMA(I)
2003    FORMAT(I3,3F)
30      CONTINUE
        CLOSE(UNIT=22)
40.     GDAT=1
        IF(IGB.EQ.1)GO TO 48

C INITIALIZING GENERATOR AND Q CONSTRAINTS
25      IF(QDAT.EQ.1)GO TO 56
        IF(TGDAT.EQ.'N')GO TO 50

C READ CONSTRAINT DATA FROM TTY/CRT
        TYPE 1006
1006    FORMAT(/1X,'**GENERATOR CONSTRAINT DATA**'/1X,'>>ENTER FOR
        1TYPE  -1 & -3  BUS (BUS #,PGMAX,PGMIN)'/9X,'FOR TYPE  -2 BUS
        1         (BUS #,PGMAX,PGMIN,QGMAX,QGMIN)'/9X,'FOR TYPE  +2 BUS
        1       (BUS #,QGMAX,QGMIN)')
        DO 36 I=1,MBUS
35      IF(BTYPE(I).GE.0)GO TO 38
        IF(BTYPE(I).EQ.-2)GO TO 37
        ACCEPT 2005,K,PGMAX(I),PGMIN(I)
2005    FORMAT(I,2F)
        GO TO 39
37      ACCEPT 2020,K,PGMAX(I),PGMIN(I),QGMAX(I),QGMIN(I)
2020    FORMAT(I,4F)
        GO TO 39
38      IF(BTYPE(I).NE.2)GO TO 36
        ACCEPT 2005,K,QGMAX(I),QGMIN(I)
39      IF(K.EQ.I)GO TO 36
        TYPE 1015,I
1015    FORMAT(1X,'WRONG ENTRY,BUS # SHOULD BE=',I3,',TRY AGAIN.')
        GO TO 35
36      CONTINUE

C ENTER CONSTRAINT DATA INTO SOS FILE FOR23.DAT
48      OPEN(UNIT=23,DEVICE='DSK',ACCESS='SEQOUT',DISPOSE='SAVE',
        1FILE='FOR23.DAT')

        DO 52 I=1,MBUS
        IF(BTYPE(I).GE.0)GO TO 53
        IF  (BTYPE(I).EQ.-2)GO TO 51
        WRITE(23,2007)I,PGMAX(I),PGMIN(I)
2007    FORMAT(1X,I3,2F11.4)
        GO TO 52
51      WRITE(23,2006)I,PGMAX(I),PGMIN(I),QGMAX(I),QGMIN(I)
2006    FORMAT(1X,I3,4F11.4)
        GO TO 52
53      IF  (BTYPE(I).NE.2)GO TO 52
        WRITE(23,2009)I,QGMAX(I),QGMIN(I)
2009    FORMAT(1X,I3,22X,2F11.4)
52      CONTINUE
        CLOSE(UNIT=23)
        IF(IGB.EQ.1)GO TO 80
        GO TO 54
C READ CONSTRAINT DATA FROM SOS FILE FOR23.DAT
50      OPEN(UNIT=23,DEVICE='DSK',ACCESS='SEQIN',DISPOSE='SAVE',
        1FILE='FOR23.DAT')
        DC 46 I=1,MBUS
        IF(BTYPE(I).GE.0)GO TO 42
        IF(BTYPE(I).EQ.-2)GO TO 41
        READ(23,*)K,PGMAX(I),PGMIN(I)
        GO TO 46
41      READ(23,2020)K,PGMAX(I),PGMIN(I),QGMAX(I),QGMIN(I)
        GO TO 46
42      IF(BTYPE(I).NE.2)GO TO 46
        READ(23,2005)K,QGMAX(I),QGMIN(I)
46      CONTINUE
        CLOSE(UNIT=23)
54      QDAT=1

C CHECK FOR PRINTOUT OF GENERATOR DATA
56      IF(CG.EQ.'N')GO TO 70
        IF(KINV.EQ.0)GO TO 61
        TYPE 1020,GNO
1020    FORMAT(/1X,'**SYSTEM GENERATOR DATA**'/1X,'NUMBER
        1OF GENERATORS =',I3/1X,'BUS#',4X,
        1'ALPHA',4X,'BETA',6X,'GAMMA')
        DO 60 I=1,MBUS
        IF((BTYPE(I).GE.0))GO TO 60
        TYPE 1030,I,ALPHA(I),BETA(I),GAMMA(I)
1030    FORMAT(1X,I3,3F10.5)
60      CONTINUE

C       PRINT GENERATOR CONSTRAINTS
61      WRITE(5,1032)
1032    FORMAT(/1X,'**GENERATOR CONSTRAINTS**'/1X,'BUS #',
        13X,'PGMAX',5X,'PGMIN',7X,'QGMAX',6X,'QGMIN')
        DO 65 I=1,MBUS
        IF((BTYPE(I)).GE.0)GO TO 64
        IF(BTYPE(I).EQ.-2)GO TO 62
        WRITE(5,1034)I,PGMAX(I),PGMIN(I)
1034    FORMAT(1X,I3,2F11.4)
        GO TO 65
62      WRITE(5,1036)I,PGMAX(I),PGMIN(I),QGMAX(I),QGMIN(I)
1036    FORMAT(1X,I3,4F11.4)
        GO TO 65
64      IF(BTYPE(I).NE.2)GO TO 65
        WRITE(5,1038)I,QGMAX(I),QGMIN(I)
1038    FORMAT(1X,I3,22X,2F11.4)
65      CONTINUE
66      CG='N'
        IF(IGR.EQ.1)GO TO 80

C CHECK FOR GENERATOR DATA CHANGES
70      IF(CHG.EQ.'N')GO TO 180
        TYPE 1040
1040    FORMAT(/1X,'**GENERATOR DATA CHANGES**'/
        11X,'>>ENTER (VARIABLE #, BUS #, NEW VALUE THEN "CR")'/
        110X,'1=ALPHA'/10X,'2=BETA'/10X,'3=GAMMA'/10X,'4=PGMAX'/10X,'
        25=PGMIN'/10X,'6=QGMAX'/10X,'7=QGMIN'/10X,'8=RETYPE ALL
        1GENERATOR DATA'/10X,'9=UPDATE SOS FILE WITH CHANGES'
        1/10X,'>>ENTER 0 THEN "CR" WHEN CHANGES COMPLETED.')
80      READ(5,2010)NAME,I,X
2010    FORMAT(2I,F)
```

```
        IF(NAME.EQ.0)GO TO 180
        IF(NAME.EQ.8)GO TO 172
        IF(NAME.EQ.9)GO TO 175
        IF(((NAME.GE.1).AND.(NAME.LE.7)).AND.((I.GE.0).AND.
        1(I.LE.MBUS)))GO TO 90
        TYPE 1050,MBUS
1050    FORMAT(1X,'WRONG ENTRY, VARIABLE MUST BE 1-8,'/
        11X,'BUS # MUST BE 0 TO',I3,',CHECK BUS TYPE IS'/
        11X,'NEGATIVE,AND TRY AGAIN.')
        GO TO 80
90      CONTINUE
100     GO TO(110,120,130,140,150,160,170)NAME
110     ALPHA(I)=X
        TYPE 1060,I,ALPHA(I)
1060    FORMAT(1X,'ALPHA(BUS #',I3,')=',E12.5)
        GO TO 80
120     BETA(I)=X
        TYPE 1070,I,BETA(I)
1070    FORMAT(1X,'BETA(BUS #',I3,')='E12.5)
        GO TO 80
130     GAMMA(I)=X
        TYPE 1080,I,GAMMA(I)
1080    FORMAT(1X,'GAMMA(BUS #',I3,')=',E12.5)
        GO TO 80
140     PGMAX(I)=X
        TYPE 1090,I,PGMAX(I)
1090    FORMAT(1X,'PGMAX(BUS #',I3,')=',E12.5)
        GO TO 80
150     PGMIN(I)=X
        TYPE 1100,I,PGMIN(I)
1100    FORMAT(1X,'PGMIN(BUS #',I3,')=',E12.5)
        GO TO 80
160     QGMAX(I)=X
        IF(QGMAX(I).LT.QG(I))QG(I)=QGMAX(I)
        TYPE 1110,I,QGMAX(I)
1110    FORMAT(1X,'QGMAX(BUS #',I3,')=',E12.5)
        GO TO 80
170     QGMIN(I)=X
        IF(QGMIN(I).GT.QG(I))QG(I)=QGMIN(I)
        TYPE 1120,I,QGMIN(I)
1120    FORMAT(1X,'QGMIN(BUS #',I3,')=',E12.5)

        GO TO 80
172     CG='Y'
        IGB=1
        GO TO 56

175     IGB=1
        IF(KINV.EQ.0)GO TO 48
        GO TO 20


180     IGB=0
        CHG='N'
        RETURN
        END


        SUBROUTINE YBUSM
C SUBROUTINE TO CALCULATE YBUS AND ZBUS MATRIX
```

```
        REAL LNGTH
        INTEGER BUSM,CMY,CMZ,EB,MBUS,SB
        DOUBLE PRECISION AJ1,AJ2,AJ3,AJ4,BJ1,BJ2,BJ3,BJ4,
        1CALP,CALQ,DQ,DVANG,DVMAG,G,H,ITL,P,Q,VANG,VMAG,
        2XX,YANG,YMAG,YY,YYSHT,ZANG,ZMAG,YSER,D

        COMMON/ /        AJ1(50,50),    AJ2(50,50),     AJ3(50,50),
        1AJ4(50,50),     ALPHA(50),     BDAT,           BETA(50),
        1BGSOL,          BJ1(50,50),    BJ2(50,50),     BJ3(50,50),
        1BJ4(50,50),     BTYPE(50),     BUSM,           CALP(50),
        1CALQ(50),       CB,CC,            CG,             CHB,
        1CHG,            CHL,           CL,             CMY,
        1CMZ,            DQ(50),        DVANG(50),      DVMAG(50),
        1EB(100),         ESP1,          ESP2,           G(50),
        1GAMMA(50),      GDAT,          H(50),          IC(50),
        1II,             ITER,          ITL(50),        KINV,
        1LAMBDA,         LDAT,          LITER,
        1LNGTH(100),      LNO,           LOSS,           LPSOL,
        1LSET,           MBUS,          NS,             NSWGB,
        1P(50),          PD(50),        PG(50),         PGMAX(50),
        1PGMIN(50),      Q(50),         QD(50),QDAT,         QG(50),
        1QGMAX(50),      QGMIN(50),     SB(100),         YYSHT(100,2),
        1TCL,TLDAT,      TPD,TBDAT,     TPL,TGDAT,      VANG(50),
        1VMAG(50),       XX(50,50),     YANG(50,50),
        1YMAG(50,50),    YSHT(100,2),    YY(50,50),      ZANG(50,50),
        1ZMAG(50,50),    ZSER(100,2),    YSER(100,2),    ABORT

        IF((BUSM.EQ.2).OR.((KINV.EQ.0).AND.(BUSM.EQ.1)))GO TO 90
        TYPE 1000
        OPEN(UNIT=24,DEVICE='DSK',ACCESS='SEQOUT',DISPOSE='SAVE',
        1FILE='FOR24.DAT')

1000    FORMAT(/1X,'**GENERATING YBUS MATRIX**')

C CALCULATING LINE SHUNT AND SERIES ADMITTANCE
        DO 15 I=1,LNO
        YYSHT(I,1)=(DBLE(YSHT(I,1))*DBLE(LNGTH(I)))/2.0D0
        YYSHT(I,2)=(DBLE(YSHT(I,2))*DBLE(LNGTH(I)))/2.0D0
        D=(DBLE(LNGTH(I)))*((DBLE(ZSER(I,1))*DBLE(ZSER(I,1)))
        1+(DBLE(ZSER(I,2))*DBLE(ZSER(I,2))))
        IF(D.EQ.0.0D0)GO TO 20
        YSER(I,1)=DBLE(ZSER(I,1))/D
        YSER(I,2)=-DBLE(ZSER(I,2))/D
        GO TO 15
20      YSER(I,1)=0.0D0
        YSER(I,2)=0.0D0
15      continue
        DO 1 I=1,LNO
1       WRITE(24,2000)YYSHT(I,1),YYSHT(I,2),YSER(I,1),YSER(I,2)
2000    FORMAT(4D)

C CALCULATING YBUS MATRIX
        DO 30 I=1,MBUS
        DO 30 J=1,MBUS
        YMAG(I,J)=0.0D0
        YANG(I,J)=0.0D0
        AJ1(I,J)=0.0D0
        AJ2(I,J)=0.0D0
        IF(J.EQ.I)GO TO 60
```

```
C OFF DIAGONAL ELEMENTS(I.NE.J)
        DO 40 L=1,LNO
        IF(((SB(L).EQ.I).AND.(EB(L).EQ.J)).OR.((SB(L).EQ.J)
      . 1.AND.(EB(L).EQ.I)))GO TO 42
        GO TO 40
42      YMAG(I,J)=DSQRT((YSER(L,1)*YSER(L,1))+(YSER(L,2)*YSER(L,2)))
        IF(YSER(L,1).EQ.(0.0D0))GO TO 22
        YANG(I,J)=DATAN2(-YSER(L,2),-YSER(L,1))
        GO TO 29
22      IF(YSER(L,2))24,26,28
24      YANG(I,J)=1.570796326794897
        GO TO 29
26      YANG(I,J)=0.0D0
        GO TO 29
28      YANG(I,J)=-1.570796326794897
29      IF(KINV.EQ.0)GO TO 40
        AJ1(I,J)=-YSER(L,1)
        AJ2(I,J)=-YSER(L,2)
40      CONTINUE
        GO TO 30

C ON DIAGONAL ELEMENTS(I.EQ.J)
60      DO 45 L=1,LNO
        IF((SB(L).EQ.I).OR.(EB(L).EQ.I))GO TO 46
        GO TO 45
46      AJ1(I,J)=AJ1(I,J)+YYSHT(L,1)+YSER(L,1)
        AJ2(I,J)=AJ2(I,J)+YYSHT(L,2)+YSER(L,2)
45      CONTINUE
        YMAG(I,J)=DSQRT(AJ1(I,J)*AJ1(I,J)+AJ2(I,J)*AJ2(I,J))
        IF(AJ1(I,J).EQ.(0.0D0))GO TO 31
        YANG(I,J)=DATAN2(AJ2(I,J),AJ1(I,J))
        GO TO 30
31      IF(AJ2(I,J))32,33,34
32      YANG(I,J)=-1.570796326794897
        GO TO 30
33      YANG(I,J)=0.0D0
        GO TO 30
34      YANG(I,J)=+1.570796326794897
30      CONTINUE
        DO 2 I=1,MBUS
2       WRITE(24,2010) (YMAG(I,J),YANG(I,J),J=1,MBUS)
2010    FORMAT(2D)
35      BUSM=1
        IF(KINV.EQ.0)GO TO 90

C CALCULATING ZBUS MATRIX
        TYPE 1005
1005    FORMAT(15X,'**CALCULATING ZBUS MATRIX**')
        DO 50 I=1,MBUS
        DO 50 J=1,MBUS
        BJ1(I,J)=AJ1(I,J)
50      BJ2(I,J)=AJ2(I,J)
        BJ3(I,J)=0.0D0
        BJ4(I,J)=0.0D0
        CALL CTRI(BJ1,BJ2,BJ3,BJ4,MBUS)
        CALL CTRIV(BJ3,BJ4,BJ1,BJ2,MBUS)
        DO 80 I=1,MBUS
        DO 80 J=1,MBUS
        ZMAG(I,J)=DSQRT(BJ1(I,J)*BJ1(I,J)+BJ2(I,J)*BJ2(I,J))
        IF(BJ1(I,J).EQ.0.0D0)GO TO 82
80      ZANG(I,J)=DATAN2(BJ2(I,J),BJ1(I,J))
```

```
        GO TO 81
82      IF(BJ2(I,J))83,84,85
83      ZANG(I,J)=-1.570796326794897
        GO TO 81
84      ZANG(I,J)=0.0D0
        GO TO 81
85      ZANG(I,J)=1.570796326794897
81      CONTINUE
        DO 4 I=1,MBUS
4       WRITE(24,2010) (ZMAG(I,J),ZANG(I,J),J=1,MBUS)
        BUSM=2

C PRINTOUT OF YBUS AND ZBUS MATRIX

90      CLOSE(UNIT=24)
        IF(CMY.EQ.'N')GO TO 105
        TYPE 4000
4000    FORMAT(/1X,'**CALCULATED SYSTEM LINE DATA**'/1X,
        1'LINE #',15X,'Y SHUNT',23X,'Y SERIES')
        DO 95 I=1,LNO
        TYPE 4100,I,YYSHT(I,1),YYSHT(I,2),YSER(I,1),YSER(I,2)
4100    FORMAT(I4,4X,2E14.6,2X,2E14.6)
95      CONTINUE
        TYPE 1010
1010    FORMAT(/1X,'**YBUS MATRIX**')
        DO 100 I=1,MBUS
        TYPE 1020,I
1020    FORMAT(1X,'ROW',I3)
        TYPE 1030,(YMAG(I,J),YANG(I,J),J=1,MBUS)
1030    FORMAT(6E12.5)
100     CONTINUE
        CMY='N'
105     IF(KINV.EQ.0)GO TO 120
        IF(CMZ.EQ.'N')GO TO 120
        TYPE 1040
1040    FORMAT(/1X,'**ZBUS MATRIX**')
        DO 110 I=1,MBUS
        TYPE 1020,I
        TYPE 1030,(ZMAG(I,J),ZANG(I,J),J=1,MBUS)
110     CONTINUE
        CMZ='N'

120     RETURN
        END


        SUBROUTINE NRLF
C        CALCULATES P,Q,V,AND ANGLE AT EACH BUS
        INTEGER ABORT,BTYPE, MBUS
        DIMENSION DP(50)

        DOUBLE PRECISION AJ1,AJ2,AJ3,AJ4,BJ1,BJ2,BJ3,BJ4,
        1CALP,CALQ,DQ,DVANG,DVMAG,G,H,ITL,P,Q,VANG,VMAG,
        2XX,YANG,YMAG,YY,YYSHT,ZANG,ZMAG,YSER,GMAX,HMAX,DP

        COMMON/ /       AJ1(50,50),    AJ2(50,50),    AJ3(50,50),
        1AJ4(50,50),    ALPHA(50),     BDAT,          BETA(50),
        1BGSOL,         BJ1(50,50),    BJ2(50,50),    BJ3(50,50),
        1BJ4(50,50),    BTYPE(50),     BUSM,          CALP(50),
        1CALQ(50),      CB,CC,           CG,            CMB,
        1CMG,           CML,            CL,             CMY,
        1CMZ,           DQ(50),         DVANG(50),      DVMAG(50),
        1EB(100),        ESP1,           ESP2,           G(50),
        1GAMMA(50),     GDAT,           H(50),          IC(50),
        1II,            ITER,           ITL(50),        KINV,
        1LAMBDA,        LDAT,           LITER,
        1LNGTH(100),     LNO,            LOSS,           LPSOL,
        1LSET,          NBUS,           N5,             NSWGB,
        1P(50),         PD(50),         PG(50),         PGMAX(50),
        1PGMIN(50),     Q(50),          QD(50),QDAT,        QG(50),
        1QGMAX(50),     QGMIN(50),      SB(100),         YYSHT(100,2),
        1TOL,TLDAT,     TPD,TBDAT,      TPL,TGDAT,      VANG(50),
        1VMAG(50),      XX(50,50),      YANG(50,50),
        1YMAG(50,50),   YSHT(100,2),     YY(50,50),      ZANG(50,50),
        1ZMAG(50,50),   ZSER(100,2),     YSER(100,2),    ABORT,IBT(50)

        LL=0
        TYPE 100
100     FORMAT(/1X,'**SOLVING LOAD FLOW EQUATIONS BY  NR**')

C       CALCULATING G AND H AT EACH BUS
        IF(LSET.EQ.1)GO TO 13
        DO 1 I=1,NBUS
        P(I)=DBLE(PG(I))-DBLE(PD(I))
        Q(I)=DBLE(QG(I))-DBLE(QD(I))
1       CONTINUE
13      CONTINUE
        DO 7 I=1,NBUS
        CALP(I)=0.0D0
        CALQ(I)=0.0D0

        DO 4 J=1,NBUS
        CALP(I)=CALP(I)+VMAG(I)*VMAG(J)*YMAG(I,J)
        1*DCOS(YANG(I,J)-VANG(I)+VANG(J))
        CALQ(I)=CALQ(I)-VMAG(I)*VMAG(J)*YMAG(I,J)
        1*DSIN(+YANG(I,J)-VANG(I)+VANG(J))
4       CONTINUE

        G(I)=P(I)-CALP(I)
        H(I)=Q(I)-CALQ(I)


7       CONTINUE
C       CHECK FOR CONVERGENCE(GMAX &HMAX.LT.TOL)

        GMAX=0.0D0
        HMAX=0.0D0

        DO 5 I=1,NBUS
        IF(GMAX.LT.DABS(G(I))) GMAX=DABS(G(I))
        IF(HMAX.LT.DABS(H(I))) HMAX=DABS(H(I))
5       CONTINUE
        IF((GMAX.LE.TOL).AND.(HMAX.LE.TOL))GO TO 56

C**********GENERATING JACOBIAN AJ1
        DO 6 I=1,NBUS
        DO 8 J=1,NBUS
        IF(IABS (BTYPE(J)).EQ.3) GO TO 10
        IF(J.EQ.I) GO TO 9
        AJ1(I,J)=-VMAG(I)*VMAG(J)*YMAG(I,J)
```

```
          1*DSIN(YANG(I,J)-VANG(I)+VANG(J))
          GO TO 8
 9        AJ1(I,J)=0.0D0

          DO 12 K=1,NBUS
          IF (K.EQ.I) GO TO 12
          AJ1(I,J)=AJ1(I,J)+VMAG(I)*VMAG(K)*YMAG(I,K)
          1*DSIN(YANG(I,K)-VANG(I)+VANG(K))
 12       CONTINUE

          GO TO 8
 10       IF(J.EQ.I) GO TO 11
          AJ1(I,J)=0.0D0
          GO TO 8
 11       AJ1(I,J)=-1.0D0

 8        CONTINUE
 6        CONTINUE

C**********GENERATING JACOBIAN AJ2
          DO 14 I=1,NBUS

          DO 17 J=1,NBUS
          IF(IABS (BTYPE(J)).GT.1) GO TO 18
          IF(J.EQ.I) GO TO 15
          AJ2(I,J)=VMAG(I)*YMAG(I,J)
          1*DCOS(YANG(I,J)-VANG(I)+VANG(J))
          GO TO 17

 15       AJ2(I,J)=0.0D0
          DO 19 K=1,NBUS
          IF (K.NE.I) GO TO 16
          AJ2(I,J)=AJ2(I,J)+2*VMAG(I)*YMAG(I,I)
          1*DCOS(+YANG(I,I))
          GC TO 19
 16       CCNTINUE
          AJ2(I,J)=AJ2(I,J)+VMAG(K)*YMAG(I,K)
          1*DCOS(YANG(I,K)-VANG(I)+VANG(K))
 19       CONTINUE
          GO TO 17

 18       AJ2(I,J)=0.0D0

 17       CONTINUE
 14       CONTINUE

C**********GENERATING JACOBIAN AJ3
          DO 20 I=1,NBUS

          DO 24 J=1,NBUS
          IF(IABS (BTYPE(J)).EQ.3) GO TO 22
          IF(J.EQ.I) GO TO 23
          AJ3(I,J)=-VMAG(I)*VMAG(J)*YMAG(I,J)
          1*DCOS(YANG(I,J)-VANG(I)+VANG(J))

          GO TO 24
 23       AJ3(I,J)=0.0D0

          DO 21 K=1,NBUS
          IF(K.EQ.J) GO TO 21
          AJ3(I,J)=AJ3(I,J)+VMAG(I)*VMAG(K)*YMAG(I,K)
```

```
        1*DCOS(YANG(I,K)-VANG(I)+VANG(K))
21      CONTINUE
        GO TO 24

22      AJ3(I,J)=0.0D0

24      CONTINUE
20      CONTINUE

C********** GENERATING JACOBIAN AJ4
        DO 29 I=1,NBUS

        DO 25 J=1,NBUS
        IF(IABS (BTYPE(J)).GT.1) GO TO 27
        IF(J.EQ.I) GO TO 28
        AJ4(I,J)=-VMAG(I)*YMAG(I,J)
        1*DSIN(YANG(I,J)-VANG(I)+VANG(J))
        GO TO 25

28      AJ4(I,J)=0.0D0

        DO 26 K=1,NBUS
        IF (K.NE.I) GO TO 30
        AJ4(I,J)=AJ4(I,J)-2*VMAG(I)*YMAG(I,I)
        1*DSIN(+YANG(I,I))
        GO TO 26
30      CONTINUE

        AJ4(I,J)=AJ4(I,J)-VMAG(K)*YMAG(I,K)
        1*DSIN(YANG(I,K)-VANG(I)+VANG(K))
26      CONTINUE
        GO TO 25
27      IF(J.EQ.I)GO TO 31
        AJ4(I,J)=0.0D0
        GO TO 25

31      AJ4(I,J)=-1.0D0
25       CONTINUE
29      CONTINUE

C       INVERTING JACOBIAN AJ4
        CALL MINV(AJ4,NBUS)

C***********GENERATING INVERSE SUBMATRIX BJ1,BJ2,BJ3,BJ4
C       BJ1=[AJ1-AJ2(AJ4**-1)AJ3]**-1

        CALL AMULT(XX,AJ2,AJ4,NBUS)
        CALL AMULT(YY,XX,AJ3,NBUS)
        DO 37 I=1,NBUS

        DO 39 J=1,NBUS
        BJ1(I,J) =AJ1(I,J)-YY(I,J)
39      CONTINUE
37      CONTINUE

C       INVERTING JACOBIAN BJ1
        CALL MINV(BJ1,NBUS)

C***********GENERATING JACOBIAN BJ2
```

```
C       BJ2=-BJ1*AJ2*(AJ4**-1)

        CALL AMULT(YY,BJ1,AJ2,MBUS)
        CALL AMULT (BJ2,YY,AJ4,MBUS)
        DO 44 I=1,MBUS
        DO 45 J=1,MBUS
        BJ2(I,J)=-1.0D0*(BJ2(I,J))
45      CONTINUE
44      CONTINUE

C**********GENERATING JACOBIAN BJ3
C       BJ3=-(AJ4**-1)*AJ3*BJ1
        CALL AMULT(YY,AJ3,BJ1,MBUS)
        CALL AMULT(BJ3,AJ4,YY,MBUS)

        DO 47 I=1,MBUS

        DO 48 J=1,MBUS
        BJ3(I,J)=-1.0D0*BJ3(I,J)

48      CONTINUE
47      CONTINUE

C**********GENERATING JACOBIAN BJ4
C       BJ4=(AJ4**-1)-(AJ4**-1)*AJ3*BJ2
        CALL AMULT(YY,AJ4,AJ3,MBUS)
        CALL AMULT(XX,YY,BJ2,MBUS)
        DO 51 I=1,MBUS
        DO 50 J=1,MBUS
        BJ4(I,J)=AJ4(I,J)-XX(I,J)
50      CONTINUE
51      CONTINUE

C       CALCULATING DELTA P,Q,VMAG,ANGLE
        DO 52 I=1,MBUS
        DP(I)=0.0D0
        DQ(I)=0.0D0
        DVMAG(I)=0.0D0
        DVANG(I)=0.0D0
        IF(IABS(BTYPE(I)).EQ.3)GO TO 54
        IF(IABS(BTYPE(I)).EQ.2)GO TO 55
        DO 53 J=1,MBUS
        DVMAG(I)=DVMAG(I)+BJ3(I,J)*G(J)+BJ4(I,J)*H(J)
53      DVANG(I)=DVANG(I)+BJ1(I,J)*G(J)+BJ2(I,J)*H(J)
        GC TO 52
54      DO 64 J=1,MBUS
        DP(I)=DP(I)+BJ1(I,J)*G(J)+BJ2(I,J)*H(J)

64      DQ(I)=DQ(I)+BJ3(I,J)*G(J)+BJ4(I,J)*H(J)
        GO TO 52
55      DO 65 J=1,MBUS
        DQ(I)=DQ(I)+BJ3(I,J)*G(J)+BJ4(I,J)*H(J)
65      DVANG(I)=DVANG(I)+BJ1(I,J)*G(J)+BJ2(I,J)*B(J)
52      CONTINUE
C       CALCULATING NEW P,Q,V,&ANGLE
        DO 62 I=1,MBUS
        P(I)=P(I)+DP(I)
        Q(I)=Q(I)+DQ(I)
        VMAG(I)=VMAG(I)+DVMAG(I)
        VANG(I)=VANG(I)+DVANG(I)
        IF(IABS(BTYPE(I)).NE.2)GO TO 62
        QMAX=QGMAX(I)-QD(I)
        QMIN=QGMIN(I)-QD(I)
        IF(Q(I).LT.DBLE(QMAX))GO TO 58
        Q(I)=DBLE(QMAX)
        BTYPE(I)=ISIGN(1,BTYPE(I))
        IBT(I)=2
        TYPE 3000,I
3000    FORMAT(1X,'BUS #',I3,2X,'EXCEEDS QGMAX,
        1BUS CHANGED TO TYPE 1')
        GO TO 62
58      IF(Q(I).GT.DBLE(QMIN)) GO TO 62
        Q(I)=DBLE(QMIN)
        BTYPE(I)=ISIGN(1,BTYPE(I))
        IBT(I)=2
        TYPE 3001,I
3001    FORMAT(1X,'BUS #',I3,2X,'EXCEEDS QGMIN,
        1BUS CHANGED TO TYPE 1')
62      CONTINUE
        LL=LL+1
        IF(LL.LT.LITER)GO TO 13

        TYPE 2300,LL
2300    FORMAT(1X,'****WILL NCT CONVERGE IN',I3,2X,' ITERATIONS')
        ABORT=1
        GO TO 60

C       PRINT OUT SOLUTIONS
56      TYPE 2400,LL
2400    FORMAT(1X,'****NR CONVERGENCE IN',I3,' ITERATIONS')
        DO 70 I=1,MBUS
        PG(I)=P(I)+PD(I)
70      QG(I)=Q(I)+QD(I)
60      RETURN
        END


        SUBROUTINE OPT
C SUBROUTINE TO CALCULATE ITL,IC, AND POWER FOR EACH GENERATOR
C CALCULATES OPTIMUM LAMBDA FOR SYSTEM

        REAL IC,LAMBDA
        INTEGER ABORT,BTYPE,MBUS
        DIMENSION PGSAV(50)

        DOUBLE PRECISION AJ1,AJ2,AJ3,AJ4,BJ1,BJ2,BJ3,BJ4,
        1CALP,CALQ,DQ,DVANG,DVMAG,G,H,ITL,P,Q,VANG,VMAG,
        2XX,YANG,YMAG,YY,YYSHT,ZANG,ZMAG,YSER,A,B
        COMMON/ /       AJ1(50,50),     AJ2(50,50),     AJ3(50,50),
        1AJ4(50,50),    ALPHA(50),      BDAT,           BETA(50),
        1BGSOL,         BJ1(50,50),     BJ2(50,50),     BJ3(50,50),
        1BJ4(50,50),    BTYPE(50),      BUSM,           CALP(50),
        1CALQ(50),      CB,CC,            CG,             CHB,
        1CHG,           CHL,            CL,             CHY,
        1CHZ,           DQ(50),         DVANG(50),      DVMAG(50),
        1EB(100),        ESP1,           ESP2,           G(50),
        1GAMMA(50),     GDAT,           H(50),          IC(50),
        1II,            ITER,           ITL(50),        KINV,
        1LAMBDA,        LDAT,           LITER,
        1LNGTH(100),    LNO,          LOSS,        LPSOL,
        1LSET,          MBUS,         N5,          NSUGB,
        1P(50),         PD(50),       PG(50),      PGMAX(50),
        1PGMIN(50),     Q(50),        QD(50),QDAT,     QG(50),
        1QGMAX(50),     QGMIN(50),    SB(100),      YYSHT(100,2),
        1TOL,TLDAT,     TPD,TBDAT,    TPL,TGDAT,   VANG(50),
        1VMAG(50),      XX(50,50),    YANG(50,50),
        1YMAG(50,50),   YSHT(100,2),   YY(50,50),    ZANG(50,50),
        1ZMAG(50,50),   ZSER(100,2),   YSER(100,2),   ABORT,IBT(50)

        II=0
5       L=0
        TPD=0.0
        TPL=0.0
        TPGCC=0.0

C COMPUTE TOTAL POWER DEMAND & LCSS AND TOTAL POWER
C GENERATED AT CONSTANT COST
        DO 10 I=1,MBUS
        TPD=TPD+PD(I)
        TPL=TPL+P(I)
        IF(BTYPE(I).GE.0)GO TO 10
        IF((ALPHA(I).EQ.0.0).AND.(BETA(I).EQ.0.0).AND.
        1(GAMMA(I).EQ.0.0).EQ.0.0)TPGCC=TPGCC+PGMAX(I)
10      CONTINUE

C CHECK FOR NEGATIVE TOTAL POWER LOSS
        IF(TPL.GE.0.0)GO TO 20
        TYPE 1000
1000    FORMAT(11X,'SYSTEM POWER LOSS IS NEGATIVE.'/11X,' CHECK
        1TOTAL SYSTEM POWER GENERATION')
        RETURN

20      IF(LOSS.EQ.'Y')TPL=0.0
        IF(TPGCC.LT.(TPD+TPL))GO TO 30
        TYPE 1010
1010    FORMAT(11X,'MAX CONSTANT COST POWER NOT USED.'/11X,
        1'LOAD FLOW PROBLEM CNLY')
        RETURN

C DETERMINE INITIAL LAMBDA
30      IF(LSET.EQ.1)GO TO 40
        LAMBDA=5000.0
        DO 50 I=1,MBUS
        IF((BTYPE(I).GE.0).OR.(ALPHA(I).EQ.0))GO TO 50
        IF(ALPHA(I).LT.LAMBDA)LAMBDA=ALPHA(I)
        PGSAV(I)=PG(I)
50      CONTINUE
        DLAM=0.8*LAMBDA
        LSET=1
40      TYPE 1020
1020    FORMAT(5X,'***OPTIMIZING COST FUNCTIONS***')

C SKIP ITL CALCULATIONS IF LOSSLESS SYSTEM
        IF(LOSS.EQ.'Y')GO TO 60

C CALCULATING ITL'S
        DO 70 I=1,MBUS
        IF(BTYPE(I).GE.0)GO TO 70
        ITL(I)=0.0D0
        DO 80 J=1,MBUS
```

```
         IF(BTYPE(J).GE.0)GO TO 80
         A=((ZMAG(I,J)*DCOS(ZANG(I,J)))/(VMAG(I)*VMAG(J)))*
         1DCOS(VANG(I)-VANG(J))
         B=((ZMAG(I,J)*DCOS(ZANG(I,J)))/(VMAG(I)*VMAG(J)))*
         1DSIN(VANG(I)-VANG(J))
         ITL(I)=ITL(I)+(P(J)*A-Q(J)*B)
80       CONTINUE
         ITL(I)=ITL(I)+ITL(I)
70       CONTINUE

C OPTIMIZING LAMBDA
60       DPSAV=0.0
65       CONTINUE
         TPG=0.0
         DO 90 I=1,NBOS
         IF(BTYPE(I).GE.0)GO TO 90
         IF((ALPHA(I).EQ.0.0).AND.(BETA(I).EQ.0.0).AND.
         1(GAMMA(I).EQ.0.0))GO TO 100
         IF(LOSS.EQ.'N')GO TO 110
         IC(I)=LAMBDA
         GO TO 120
110      IC(I)=LAMBDA*(1.0-ITL(I))
         IF(IC(I).GE.ALPHA(I))GO TO 120
         PG(I)=0.0
         GO TO 89

C COMPOTE PG'S FROM COST FUNCTIONS
120      IF(GAMMA(I).NE.0.0)GO TO 130
         PG(I)=(IC(I)-ALPHA(I))/BETA(I)
         GO TO 89
130      C=BETA(I)*BETA(I)-4.0*(ALPHA(I)-IC(I))*GAMMA(I)
         IF(C.GE.0.0)GO TO 140
C        TYPE 1030,I
1030     FORMAT(1X,'COST CURVE COEFFICIENTS GIVE NEGATIVE
         1DETERMINANT,GENERATOR',I3,2X,'SET TO ZERO')
135      PG(I)=0.0
         GO TO 89
140      C=SQRT(C)
         PG(I)=(-BETA(I)+C)/(2.0*GAMMA(I))
         IF(PG(I).GE.PGMIN(I))GO TO 89
         PG(I)=(-BETA(I)-C)/(2.0*GAMMA(I))
         IF(PG(I).GE.PGMIN(I))GO TO 89
         PG(I)=PGMIN(I)
C        TYPE 1032,I
1032     FORMAT(1X,I3,1X,'PG NEG, GEN. SET TO 0.0')
         GO TO 89
100      PG(I)=PGMAX(I)
         IF((ALPHA(I).EQ.0.0).AND.(BETA(I).EQ.0.0).AND.
         1(GAMMA(I).EQ.0.0))GO TO 150
89       CONTINUE

C CHECK FOR PGMAX CONSTRAINTS
         IF(PG(I).LE.PGMAX(I))GO TO 105
         PG(I)=PGMAX(I)
C        TYPE 1036,I,PGMAX(I)
1036     FORMAT(1X,I3,'PG EXCEEDS PGMAI, GEN RESET TC',F12.7)
         GO TO 165
105      IF(PG(I).GE.PGMIN(I))GO TO 150
         PG(I)=PGMIN(I)
C        TYPE 1038,I,PGMIN(I)
1038     FORMAT(1X,I3,1X,'PG EXCEEDS PGMIN, GEN RESET TO';F12.5)
```

```
165       IF(GAMMA(I).NE.0.0)GO TO 160
          IC(I)=ALPHA(I)+BETA(I)*PG(I)
          GO TO 150
160       IC(I)=ALPHA(I)+BETA(I)*PG(I)+GAMMA(I)*PG(I)*PG(I)
150       TPG=TPG+PG(I)
90        CONTINUE

C CHECK POWER BALANCE LESS THAN ESP1
          DP=TPG-TPD-TPL
          IF(ABS(DP).LE.ESP1)GO TO 210
          L=L+1
          IF(L.LE.ITER)GO TO 170
          TYPE 1040,L
1040      FORMAT(11X,'POWER BALANCE EQUATIONS NOT SATISFIED IN'
          1,I3,2X,'ITERATIONS,'/11X,'CHECK COST CURVES AND
          2TOLERANCES')
          RETURN

C ADJUST LAMBDA IN COMPARISON TO PREVIOUS ITERATION
170       IF((DPSAV.GE.0.0).AND.(DP.GT.0.0))GO TO 180
          IF((DPSAV.LE.0.0).AND.(DP.LT.0.0))GO TO 190
          IF((DPSAV.GE.0.0).AND.(DP.LT.0.0))GO TO 200
          DLAM=0.5*DLAM
180       LAMBDA=LAMBDA-DLAM
          DPSAV=DP
          GO TO 85
200       DLAM=0.5*DLAM
190       LAMBDA=LAMBDA+DLAM
          DPSAV=DP
          GO TO 85
210       II=II+1
          TYPE 1050,L
1050      FORMAT(11X,'LAMBDA DETERMINED IN ',I3,2X,'ITERATIONS')

C CHECK PG(I) CONVERGENCE .LE. ESP2
          DO 220 I=1,NBUS
          IF(BTYPE(I).GE.0)GO TO 220
          IF(ABS(PG(I)-PGSAV(I)).GT.ESP2)GO TO 230
220       CONTINUE
          RETURN

230       DO 240 I=1,NBUS
          PGSAV(I)=PG(I)
          P(I)=DBLE(PG(I))-DBLE(PD(I))
          IF(IBT(I).NE.2)GO TO 240
          BTYPE(I)=ISIGN(2,BTYPE(I))
          VMAG(I)=1.0
          IBT(I)=0
240       CONTINUE
          DLAM=0.5*LAMBDA
250       CALL NRLF
          GO TO 5
          END


          SUBROUTINE CTRI(A,B,C,D,N)
          DOUBLE PRECISION A(50,50),B(50,50),C(50,50),D(50,50),T(2),SUM(2)

C  TRIANGULARIZE A INTO B (LOWER) AND
```

```
C     B UPPER TRIAG MATRIX
      DO 100 I=1,N
       C(I,1)=A(I,1)
100    D(I,1)=B(I,1)
      DO 105 J=2,N
      CALL CDIV(T(1),T(2),A(1,J),
       1B(1,J),A(1,1),B(1,1))
      C(1,J)=T(1)
105   D(1,J)=T(2)

      DO 275 IJ=2,N
      DO 250 J=IJ,N
      DO 250 I=IJ,N
      IF(I.LT.J) GO TO 205
      JM1=J-1
      SUM(2)=0.0D0
      SUM(1)=SUM(2)
      DO 200 K=1,JM1
      CALL CMUL(T(1),T(2),C(I,K),
     1 D(I,K),C(K,J),D(K,J))
      SUM(1)=SUM(1)+T(1)
200   SUM(2)=SUM(2)+T(2)
      C(I,J)=A(I,J)-SUM(1)
      D(I,J)=B(I,J)-SUM(2)
      GO TO 250
205   CONTINUE
      CALL CDIV(T(1),T(2),C(J,I),
     1 D(J,I),C(I,I),D(I,I))
      C(I,J)=T(1)
      D(I,J)=T(2)
250   CONTINUE
275   CONTINUE

      DO 280 I=1,N
      DO 280 J=1,N
      IF(I.GE.J) GO TO 280
      D(I,J)=0.0D0
       C(I,J)=D(I,J)
280   CONTINUE
      RETURN
      END

      SUBROUTINE CTRIV(A,B,C,D,N)
      DOUBLE PRECISION A(50,50),B(50,50),C(50,50),D(50,50),
       1SUM(2),T(2),AI

      I=N
10    CONTINUE
      K=I
20    CONTINUE
      SUM(2)=0.0D0
      SUM(1)=SUM(2)
      AI=SUM(1)
      IF(K-I) 25,22,25
22    AI=1.0D0
25    J=K+1
      IF(J.GT.N) GO TO 40

      DO 30 JJ=J,N
      CALL CMUL(T(1),T(2),A(JJ,K),
     1 B(JJ,K),C(I,JJ),D(I,JJ))
       SUM(1)=SUM(1)+T(1)
30     SUM(2)=SUM(2)+T(2)
40     T(1)=AI-SUM(1)
       T(2)=-SUM(2)
       CALL CDIV(SUM(1),SUM(2),T(1),T(2)
     1,A(K,K),B(K,K))
       C(K,I)=SUM(1)
       C(I,K)=C(K,I)
       D(K,I)=SUM(2)
       D(I,K)=D(K,I)

       K=K-1
       IF(K.EQ.0) GO TO 50
       GO TO 20
50     I=I-1
       IF(I.EQ.0) GO TO 60
       GO TO 10
60     CONTINUE

       RETURN

       END



        SUBROUTINE SOL

C SUBROUTINE TO CALCULATE AND PRINT BUS & GENERATOR SOLUTIONS
C      AND LINE POWER FLOW

        REAL LAMBDA,IC
        INTEGER BTYPE,BGSOL,EB,SB
        DOUBLE PRECISION AJ1,AJ2,AJ3,AJ4,BJ1,BJ2,BJ3,BJ4,
        1CALP,CALQ,DQ,DVANG,DVMAG,G,H,ITL,P,Q,VANG,VMAG,
        2XX,YANG,YMAG,YY,YYSHT,ZANG,ZMAG,YSER,
        3YA,YM,YSHTA,YSHTM,PLM,QLM


        COMMON/ /       AJ1(50,50),    AJ2(50,50),    AJ3(50,50),
        1AJ4(50,50),    ALPHA(50),     BDAT,          BETA(50),
        1BGSOL,         BJ1(50,50),    BJ2(50,50),    BJ3(50,50),
        1BJ4(50,50),    BTYPE(50),     BUSM,          CALP(50),
        1CALQ(50),      CB,CC,           CG,            CHB,
        1CHG,           CHL,           CL,            CMY,
        1CMZ,           DQ(50),        DVANG(50),     DVMAG(50),
        1EB(100),        ESP1,          ESP2,          G(50),
        1GAMMA(50),     GDAT,          H(50),         IC(50),
        1II,            ITER,          ITL(50),       KINV,
        1LAMBDA,        LDAT,          LITER,
        1LNGTH(100),     LNO,           LOSS,          LPSOL,
        1LSET,          MBUS,          NS,            NSWGB,
        1P(50),         PD(50),        PG(50),        PGMAX(50),
        1PGMIN(50),     Q(50),         QD(50),QDAT,       QG(50),
        1QGMAX(50),     QGMIN(50),     SB(100),        YYSHT(100,2),
        1TOL,TLDAT,     TPD,TBDAT,     TPL,TGDAT,     VANG(50),
        1VMAG(50),      XX(50,50),     YANG(50,50),
        1YMAG(50,50),   YSHT(100,2),    YY(50,50),      ZANG(50,50),
        1ZMAG(50,50),   ZSER(100,2),    YSER(100,2),      ABORT
        IF(BGSOL.EQ.0)GO TO 280
        IF(KINV.EQ.1)GO TO 265
        TPD=0.0
        TPL=0.0
        DO 275 I=1,MBUS
        TPD=TPD+PD(I)
        TPL=TPL+P(I)
275     CONTINUE
        WRITE(N5,1157)TPL,TPD
1157    FORMAT(/1X,'****SYSTEM BUS AND GENERATCR SOLUTIONS****'/
        11X,'SYSTEM LOSSES(PU)=',F8.4,',  SYSTEM LOAD(PU)=',F8.4,/
        21X,'BUS #',1X,'BTYPE',6X,'PG',6X,'CG',6X,'PD',6X,
        3'QD',5X,'VMAG',4X,'VANG')
        DO 277 I=1,MBUS
        PG(I)=P(I)+PD(I)
        QG(I)=Q(I)+QD(I)
277     WRITE(N5,1158)I,BTYPE(I),PG(I),QG(I),PD(I),QD(I),
        1VMAG(I),VANG(I)
1158    FORMAT(1X,I3,3X,I3,4X,6F8.4)
        BGSOL=0
        GO TO 280
265     WRITE(N5,1155)II,ESP1,ESP2,LAMBDA,TPL,TPD
1155    FORMAT(/1X,'****SYSTEM BUS AND GENERATOR SOLUTIONS****'/
        16X,'ECON. DISPATCH CONVERGENCE IN',I3,' ITERATIONS'/
        12X,'ESP1=',F8.4,', ESP2=',F8.4,',  LAMBDA=',F8.4,/1X,'SYSTEM
        3LOSSES(PU)=',F8.4,',  SYSTEM LOAD(PU)=',F8.4,/
        41X,'BUS#',1X,'BTYPE',4X,'IC',5X,'ITL',6X,'PG',6X,
        5'QG',6X,'PD',6X,'QD',5X,'VMAG',4X,'VANG')

        DO 270 I=1,MBUS
        PG(I)=P(I)+PD(I)
        QG(I)=Q(I)+QD(I)
270     WRITE(N5,1160)I,BTYPE(I),IC(I),ITL(I),PG(I),QG(I),
        1PD(I),QD(I),VMAG(I),VANG(I)
1160    FORMAT(1X,I3,1X,I3,F10.4,7F8.4)
        BGSOL=0
280     CONTINUE
        IF(LPSOL.EQ.0)GO TO 300
                IF(KINV.EQ.0)GO TO 287

C   CALCULATING BUS POWER COST
        B=0.0
        C=0.0
        WRITE(N5,1165)
1165    FORMAT(/1X,'****SYSTEM OPERATING COST****')
        DO 285 I=1,MBUS
        IF(BTYPE(I).GE.0)GO TO 285
        B=ALPHA(I)*PG(I)+(BETA(I)*PG(I)*PG(I))/2
        1+(GAMMA(I)*PG(I)*PG(I)*PG(I))/3
        WRITE(N5,1170)I,B
1170    FORMAT(1X,'GEN.#',I3,'=',F8.2,'P.U.')
        C=C+B
285     CONTINUE
        WRITE(N5,1180)C
1180    FORMAT(1X,'TOTAL SYSTEM OPERATING COST =',F10.2,'P.U.')

C CALCULATING LINE POWER FLOW
287     WRITE(N5,1190)
1190    FORMAT(/1X,'****LINE POWER FLOW****'/1X,'LINE#',3X,'SB',4X,'EB',
        14X,'REAL POWER',4X,'REACTIVE POWER')
        DO 300 I=1,LNO
        L=SB(I)
        M=EB(I)
        YM=DSORT((YSER(I,1)*YSER(I,1))+(YSER(I,2)*YSER(I,2)))
        YSHTM=DSQRT((YYSHT(I,1)*YYSHT(I,1))
        1+(YYSHT(I,2)*YYSHT(I,2)))
        IF(YSER(I,1).EQ.0.0D0)GO TO 302
        YA=DATAN2(YSER(I,2),YSER(I,1))
        GO TO 306
302     IF(YSER(I,2))303,304,305
303     YA=-1.570796326794897
        GO TO 306
304     YA=0.0D0
        GO TO 306
305     YA=1.570796326794897
306     IF(YYSHT(I,1).EQ.0.0D0)GO TO 308
        YSHTA=DATAN2(YYSHT(I,2),YYSHT(I,1))
        GO TO 314
308     IF(YYSHT(I,2))309,311,312
309     YSHTA=-1.570796326794897
        GO TO 314
311     YSHTA=0.0D0
        GO TO 314
312     YSHTA=1.570796326794897
314     CONTINUE

        JJ=0
310     PLM=VMAG(L)*VMAG(L)*YM*DCOS(YA)
        1-VMAG(L)*VMAG(M)*YM*DCOS(VANG(L)-VANG(M)-YA)
        2+VMAG(L)*VMAG(L)*YSHTM*DCOS(YSHTA)

        QLM=-VMAG(L)*VMAG(L)*YM*DSIN(YA)
        1-VMAG(L)*VMAG(M)*YM*DSIN(VANG(L)-VANG(M)-YA)
        2-VMAG(L)*VMAG(L)*YSHTM*DSIN(YSHTA)
        WRITE(N5,1200)I,L,M,PLM,QLM
1200    FORMAT(1X,I3,4X,I3,3X,I3,F12.4,6X,F12.4)
        IF(JJ.EQ.1)GO TO 300
        JJ=1
        L=EB(I)
        M=SB(I)
        GO TO 310
300     CONTINUE
        LPSOL=0
        RETURN
        END


        SUBROUTINE AMULT(Z,X,Y,M)
        DOUBLE PRECISION  Z(50,50),X(50,50),Y(50,50)

        DO 1 I=1,M
        DO 1 J=1,M
        Z(I,J)=0.0D0
        DO 1 K=1,M
        Z(I,J)=Z(I,J)+X(I,K)*Y(K,J)
1       CONTINUE
        RETURN
        END

        SUBROUTINE CMUL(X,Y,A,B,C,D)
```

```
      DOUBLE PRECISION X,Y,A,B,C,D
      X=A*C-B*D
      Y=A*D+B*C
      RETURN
      END
      SUBROUTINE CDIV(X,Y,A,B,C,D)
      DOUBLE PRECISION X,Y,A,B,C,D
      X=C*C+D*D
      Y=(B*C-A*D)/X
      X=(A*C+B*D)/X
      RETURN
      END


      SUBROUTINE CINVT(A,B,C,D)
      A=C/(C*C+D*D)
      B=-D/(C*C+D*D)
      RETURN
      END


      FUNCTION CABS(A,B)
      DOUBLE PRECISION CABS,A,B,DSQRT
      CABS=DSQRT(A*A+B*B)
      RETURN
      END


      SUBROUTINE DPINVT(A,B,C,D)
      DOUBLE PRECISION A,B,C,D
      A=C/(C*C+D*D)
      B=-D/(C*C+D*D)
      RETURN
      END


      SUBROUTINE MINV (A,N)
C     THIS SUBPROGRAM IS FOR MATRIX INVERSION AND SIMUL
C     LINEAR EQUATIONS
C     A=THE GIVEN COEFFICIENT MATRIX A;A**-1 WILL BE
C     STORED IN THIS MATRIX AT RETURN TO THE MAIN PROGRAM
C
C     N=ORDER OF A;N.GE.1
      DIMENSION  IPVOT(50),INDEX(50,2)
      DOUBLE PRECISION A(50,50),B(50,50),PIVOT(50),T
      EQUIVALENCE (IROW,JROW),(ICOL,JCOL)
C
C     FOLLOWING 3 STATEMENTS FOR INITIALIZATION
      M=0
57    DET=1.0
      DO 17 J=1,N
17    IPVOT(J)=0
C
      DO 135 I=1,N
C
C     FOLLOWING 12 STATEMENTS FOR SEARCH FOR
C     PIVOT ELEMENT
C
```

```
          T=0.0D0
          DO 9 J=1,N
          IF(IPVOT(J).EQ.1) GO TO 9
13        DO 23 K=1,N
          IF(IPVOT(K)-1) 43,23,81
43        IF(DABS(T).GE.DABS(A(J,K)))  GO TO 23
83        IROW=J
          ICOL=K
          T=A(J,K)
23        CONTINUE
9         CONTINUE
C
          IPVOT(ICOL)=IPVOT(ICOL)+1
C
C         FOLLOWING 15 STATEMENTS TO PUT PIVOT
C         ELEMENT ON DIAGONAL
C
          IF(IROW.EQ.ICOL) GO TO 109
73        DET = -DET
          DO 12 L=1,N
          T=A(IROW,L)
          A(IROW,L) = A(ICOL,L)
12        A(ICOL,L) = T
          IF(M.LE.0) GO TO 109
33        DO 2 L=1,M
          T=B(IROW,L)
          B(IROW,L)=B(ICOL,L)
2         B(ICOL,L)=T
109       INDEX(I,1)=IROW
          INDEX(I,2)=ICOL
          PIVOT(I)=A(ICOL,ICOL)
          DET=DET*PIVOT(I)
C
C         FOLLOWING 6 STATEMENTS TO DIVIDE PIVCT ROW
C         BY PIVOT ELEMENT
C
          A(ICOL,ICOL)=1.0D0
          DO 205 L=1,N
205       A(ICOL,L)=A(ICOL,L)/PIVOT(I)
          IF (M.LE.0) GO TO 347
66        DO 52 L=1,M
52        B(ICOL,L)=B(ICOL,L)/PIVOT(I)
C
C         FOLLOWING 10 STATMENTS TO REDUCE NON-PIVOT ROWS
C
C
347       DO 135 LI=1,N
          IF(LI.EQ.ICOL)  GO TO 135
21        T=A(LI,ICOL)
          A(LI,ICOL)=0.0D0
          DO 89 L=1,N
89        A(LI,L)=A(LI,L)-A(ICCL,L)*T
          IF(M.LE.0) GO TO 135
18        DO 68 L=1,M
68        B(LI,L)=B(LI,L)-B(ICOL,L)*T
135       CONTINUE
C
C         FOLLOWING 11 STATEMENTS TO INTERCHANGE CONLUMS
C
222       DO 3 I=1,N
          L=N-I+1
```

```
        IF(INDEX(L,1).EQ.INDEX(L,2)) GO TO 3
19      JROW = INDEX(L,1)
        JCOL = INDEX(L,2)
        DO 549 K=1,N
        T = A(K,JROW)
        A(K,JROW)=A(K,JCOL)
        A(K,JCOL)=T
549     CONTINUE
3       CONTINUE
81      RETURN
        END
```

Computer Code EDSP15

```
C ECONOMIC DISPATCH EXECUTIVE PROGRAM
C        THIS PROGRAM PROVIDES LOAD FLOW SOLUTIONS AND ECONOMIC
C        DISPATCH FOR POWER SYSTEMS WITH A MAXIMUM OF 15
C        BUSES AND 30 TRANSMISSION LINES. IT SOLVES FOR SYSTEM
C        LOAD,LOSSES,GENERATOR INCREMENTAL COSTS,REAL POWER
C        GENERATION,TRANSMISSION LINE REAL & REACTIVE POWER FLOWS.

         REAL LNGTH,LAMBDA,IC
         INTEGER ABORT,ANS,BDAT,BTYPE,BUSM,CB,CC,CG,CBC,CMB,CMG,CML,
         1CL,CMY,CMZ,EB,GDAT,QDAT,MBUS,BGSOL,LFSOL,SB,TLDAT,TBDAT,TGDAT
         DOUBLE PRECISION AJ1,AJ2,AJ3,AJ4,BJ1,BJ2,BJ3,BJ4,
         1CALP,CALQ,DQ,DVANG,DVMAG,G,H,ITL,P,Q,VANG,VMAG,
         2XX,YANG,YMAG,YY,YYSHT,ZANG,ZMAG,YSER
         COMMON/ /       AJ1(15,15),     AJ2(15,15),     AJ3(15,15),
         1AJ4(15,15),    ALPHA(15),      BDAT,           BETA(15),
         1BGSOL,         BJ1(15,15),     BJ2(15,15),     BJ3(15,15),
         1BJ4(15,15),    BTYPE(15),      BUSM,           CALP(15),
         1CALQ(15),      CB,CC,             CG,             CMB,
         1CMG,           CML,            CL,             CMY,
         1CMZ,           DQ(15),         DVANG(15),      DVMAG(15),
         1EB(30),        ESP1,           ESP2,           G(15),
         1GAMMA(15),     GDAT,           H(15),          IC(15),
         1II,            ITER,           ITL(15),        KINV,
         1KSOL,          LAMBDA,         LDAT,           LITER,
         1LNGTH(30),     LNO,            LOSS,           LPSOL,
         1LSET,          MBUS,           N5,             NSWGB,
         1P(15),         PD(15),         PG(15),         PGMAX(15),
         1PGMIN(15),     Q(15),          QD(15),QDAT,        QG(15),
         1QGMAX(15),     QGMIN(15),      SB(30),          YYSHT(30,2),
         1TOL,TLDAT,     TPD,TBDAT,      TPL,TGDAT,       VANG(15),
         1VMAG(15),      VSPEC(15),      XX(15,15),       YANG(15,15),
         1YMAG(15,15),   YSHT(30,2),     YY(15,15),       ZANG(15,15),
         1ZMAG(15,15),   ZSER(30,2),     YSER(30,2),       ABORT,IBT(15)
         DATA
         1ABORT/0/,         BDAT/0/,          BUSM/0/,          CB/'N'/,
         1CG/'N'/,            CMB/'N'/,           CBC/'N'/,
         1CMG/'N'/,           CML/'N'/,           CL/'N'/,
         1CMY/'N'/,           CMZ/'N'/,         ESP1/0.001/,
         1ESP2/0.001/,        GDAT/0/,          ITER/50/,          KINV/0/,
         1KSOL/0/,          LDAT/0/,          LITER/5/,         LOSS/'N'/,
         1TOL/0.0001/,       QDAT/0/

         CALL ERRSET(0)
10       TYPE 1000
1000     FORMAT(15X'****ECONOMIC DISPATCH EXECUTIVE PROGRAM****')

C SYSTEM INITIALIZATION
20       TYPE 1010
1010     FORMAT(/1X,'>>ENTER TYPE PROBLEM'/8X,'0=LOAD FLOW PROBLEM'/
         18X,'1=ECONOMIC DISPATCH PROBLEM')
22       ACCEPT 2000,KINV
2000     FORMAT(I1)
         IF((KINV.EQ.0).OR.(KINV.EQ.1)) GO TO 24
         TYPE 1012
1012     FORMAT(1X,'WRONG ENTRY, MUST BE 0 OR 1, TRY AGAIN')
         GO TO 22
24       IF(KINV)30,30,26
26       TYPE 1014
1014     FORMAT(/1X,'>>IS SYSTEM LOSSLESS?')
28       ACCEPT 2001,LOSS
2001     FORMAT(1A1)
```

```
        IF((LOSS.EQ.'N').OR.(LOSS.EQ.'Y'))GO TO 30
        TYPE 1013
1013    FORMAT(1X,'WRONG ENTRY, MUST BE "Y" OR "N", TRY AGAIN.')
        GO TO 28
30      TYPE 1018
1018    FORMAT(/1X,'>>ENTER LOAD FLOW SOLUTION METHOD'/8X,
        1'0=NEWTON-RAPHSON'/8X,'1=ACCELERATED GAUSS-SEIDEL'/8X,
        1'2=PATTERN SEARCH')
32      ACCEPT 2000,KSOL
        IF((KSOL.GE.0).AND.(KSOL.LE.2))GO TO 34
        TYPE 1017
1017    FORMAT(1X,'WRONG ENTRY,MUST BE 0,1, OR 2,TRY AGAIN.')
        GO TO 32
34      IF(KSOL.EQ.2)KSOL=0
        IF((KSOL.EQ.1).AND.(LITER.LT.30))LITER=30
        IF((LDAT.EQ.1).AND.(BDAT.EQ.1).AND.(GDAT.EQ.1))
        1GO TO 70
        TYPE 1020
1020    FORMAT(/1X,'>>ENTER TOTAL NUMBER OF (LINES, & BUSES).')
36      ACCEPT 2004,LNO,MBUS
2004    FORMAT(2I)
        IF(((LNO.GE.1).AND.(LNO.LE.30)).AND.((MBUS.GE.1)
        1.AND.(MBUS.LE.15)))GO TO 40
        TYPE1016
1016    FORMAT(1X,'WRONG ENTRY,TOTAL LINES=1 TO 30 &TOTAL BUSES=1 TO 15,
        1TRY AGAIN.')
        GO TO 36

C CHECK METHOD OF DATA INPUT(SOS FILE/CRT/TTY)
40      TYPE 1030
1030    FORMAT(/1X,'>>WANT TO TYPE IN NEW(LINE DATA, BUS DATA, OR
        1 GENERATOR DATA)?')
50      ACCEPT 2008,TLDAT,TBDAT,TGDAT
        IF(((TLDAT.EQ.'N').OR.(TLDAT.EQ.'Y')).AND.((TBDAT.EQ.'N')
        1.OR.(TBDAT.EQ.'Y')).AND.((TGDAT.EQ.'N').OR.(TGDAT.EQ.'Y')))
        2GO TO 60
        TYPE 1015
1015    FORMAT(1X,'WRONG ENTRY, MUST BE "N" OR "Y" FOR
        1EACH DATA GROUP,'/1x,'i.e. "NYN" means type in bus
        1data only,please use CAP N & Y,'/1x,'try again.')
2008    FORMAT(3A1)
        GO TO 50

C READ SYSTEM DATA FROM SOS FILE,TTY,OR CRT
60      IF(LDAT.EQ.0)CALL LDATA
        IF(ABORT.EQ.1)GO TO 320
80      IF(BDAT.EQ.0)CALL BDATA
        IF(ABORT.EQ.1)GO TO 320
90      IF(GDAT.EQ.0)CALL GDATA
        IF(ABORT.EQ.1)GO TO 320

C CHECK FOR PRINTOUT OF SYSTEM DATA
70      TYPE 1050
1050    FORMAT(/1X,'>>WANT A COPY OF (LDATA,BDATA,GDATA,
        1OR SYSTEM CONVERGENCE CRITERIA)?')
100     READ(5,2020)CL,CB,CG,CC
2020    FORMAT(4A1)
        IF(((CL.EQ.'N').OR.(CL.EQ.'Y')).AND.
        1((CB.EQ.'N').OR.(CB.EQ.'Y')).AND.
        2((CG.EQ.'N').OR.(CG.EQ.'Y')).AND.
        3((CC.EQ.'N').OR.(CC.EQ.'Y')))GO TO 110
        TYPE 1019
1019    FORMAT(1X,'WRONG ENTRY, MUST BE "N" OR "Y" FOR
        1EACH DATA GROUP.'/1X,'I.E."NYNY" MEANS TYPE OUT
        2BUS DATA AND CONVERGENCE CRITERIA ONLY.'/1X, 'TRY AGAIN.')
        GO TO 100
110     IF(CL.EQ.'Y')CALL LDATA
120     IF(CB.EQ.'Y')CALL BDATA
130     IF(CG.EQ.'Y')CALL GDATA

C       PRINT SYSTEM CONVERGENCE CRITERIA
140     IF(CC.EQ.'N')GO TO 145
142     WRITE(5,1065)LITER,TOL,ITER,ESP1,ESP2
1065    FORMAT(/1X,'**SYSTEM CONVERGENCE CRITERIA**'/10X,
        1'LOAD FLOW ITERATIONS=',I3/
        110X,'LOAD FLOW TOLERANCE=',E13.5/
        210X,'ECON DISP ITERATIONS=',I3/10X,'ESP1=',E13.5/10X,'ESP2=',
        3E13.5)

C CHECK FOR CHANGES TO CONVERGENCE CRITERIA OR SYSTEM DATA
145     TYPE 1070
1070    FORMAT(/1X,'>>WANT TO CHANGE
        1(LDATA,BDATA,GDATA, OR CONVERGENCE CRITERIA)?')
150     ACCEPT 2040,CHL,CHB,CHG,CHC
2040    FORMAT(4A1)
        IF(((CHC.EQ.'N').OR.(CHC.EQ.'Y')).AND.
        1((CHL.EQ.'N').OR.(CHL.EQ.'Y')).AND.
        1((CHB.EQ.'N').OR.(CHB.EQ.'Y')).AND.
        1((CHG.EQ.'N').OR.(CHG.EQ.'Y')))GO TO 160
        TYPE 1021
1021    FORMAT(1X,'WRONG ENTRY, MUST BE "N" OR "Y" FOR EACH DATA
        1GROUP, TRY AGAIN.')
        GO TO 150
160     IF(CHL.EQ.'Y')CALL LDATA
        IF(CHB.EQ.'Y')CALL BDATA
        IF(CHG.EQ.'Y')CALL GDATA
        IF(CHC.EQ.'N')GO TO 190
        TYPE 1090
1090    FORMAT(/1X,'**CONVERGENCE CRITERIA CHANGES**'/
        11x,'>>ENTER(VARIABLE #,NEW VALUE THEN "CR")'/
        110X,'1=LOAD FLOW ITERATIONS'/10X,'2=LOAD FLOW TOLERANCE'/
        210X,'3=ECON DISP ITERATIONS'/10X,'4=ESP1'/10X,'5=ESP2'/
        310X,'>>ENTER 0 THEN "CR" WHEN CHANGES COMPLETED.')
170     READ(5,2050)NAME,X
2050    FORMAT(I,F)
        IF((NAME.GE.0).AND.(NAME.LE.5))GO TO 180
        TYPE 1100
1100    FORMAT(1X,'WRONG ENTRY,VARIABLE MUST BE 1-5, TRY AGAIN.')
        GO TO 170
180     IF(NAME.EQ.0)GO TO 190
        GO TO(181,182,183,184,185)NAME
181     LITER=IFIX(X)
        TYPE 1110,LITER
1110    FORMAT(8X,'LOAD FLOW ITERATIONS=',I3)
        GO TO 170
182     TOL=X
        TYPE 1120,TOL
1120    FORMAT(8X,'LOAD FLOW TOLERANCE=',E13.5)
        GO TO 170
183     ITER=IFIX(X)
        TYPE 1130,ITER
```

```
1130      FORMAT(8X,'OPT DISP ITERATIONS=',I3)
          GO TO 170
184       ESP1=X
          TYPE 1140,ESP1
1140      FORMAT(8X,'ESP1=',E13.5)
          GO TO 170
185       ESP2=X
          TYPE 1150,ESP2
1150      FORMAT(8X,'ESP2=',E13.5)
          GO TO 170
190       IF((BUSM.EQ.0).OR.((BUSM.NE.2).AND.(KINV.EQ.1)))CALL YBUSM
222       TYPE 1152
1152      FORMAT(/1X,'>>WANT A COPY OF (YBUS MATRIX, ZBUS MATRIX)?')
223       ACCEPT 2060,CMY,CMZ
          IF(KINV.EQ.0)CMZ='N'
2060      FORMAT(2A1)
          IF(((CMY.EQ.'N').OR.(CMY.EQ.'Y')).AND.((CMZ.EQ.'N').OR.
          1(CMZ.EQ.'Y')))GO TO 224
          TYPE 1021
          GO TO 223
224       IF ((CMY.EQ.'Y').OR.(CMZ.EQ.'Y'))CALL YBUSM
225       CONTINUE

C SYSTEM INITIALIZED STARTING LOAD FLOW CALCULATIONS
          LSET=0
230       IF(KSOL) 240,240,250

C CALCULATING LOAD FLOW BY NEWTON-RAPHSON
240       CALL NRLF
          GO TO 260
C CALCULATING LOAD FLOW BY GAUSS-SEIDEL
250       CALL GSLF
260       IF(ABORT.EQ.1)GO TO 320
          IF (KINV.EQ.0)GO TO 320

C CALCULATING OPT DISP POWER BALANCE EQUATIONS
          CALL OPT
          GO TO 330

320       ABORT=0
330       TYPE1210
1210      FORMAT(/1X,'>>TYPE CODE # FOR DESIRED ACTION'/
          18X,'1=PRINT BUS & GEN SOLUTIONS ONLY.'/8X,'2=PRINT LINE
          2POWERS, BUS & GEN SOLUTIONS'/8X,'3=CREATE SOS FILE OF
          2 SOLUTIONS'/8X,'4=CHANGE DATA OR TOLERANCES AND RERUN'/
          18X,'5=CHANGE TYPE PROBLEM OR SOLUTION METHOD,'/13X,'RERUN WITH
          1EXISTING DATA'/8X,'6=NEW PROBLEM'/8X,'7=TERMINATE')

340       ACCEPT 2000,ANS
          IF((ANS.GE.1).AND.(ANS.LE.7))GO TO 350
          TYPE 1220
1220      FORMAT(1X,'WRONG ENTRY, MUST BE 1 TO 7, TRY AGAIN.')
          GO TO 340
350       GO TO(361,362,380,363,365,367,500)ANS
361       N5=5
          BGSOL=1
          LPSOL=0
          CALL SOL
          GO TO 330
362       N5=5
          BGSOL=1
```

```
        LPSOL=1
        CALL SOL
        GO TO 330
363     BDAT=0
        TBDAT='N'
        CALL BDATA
        GO TO 145
365     LSET=0
        BDAT=0
        TBDAT='N'
        CALL BDATA
        GO TO 10
367     LDAT=0
        BDAT=0
        GDAT=0
        QDAT=0
        LSET=0
        GO TO 10

C CREATING SOS FILES OF LINE, BUS, GENERATOR DATA,
C YBUS & ZBUS MATRIX, BUS & GENERATOR SOLUTIONS, AND
C LINE POWER FLOW, AND COMMAND FILE FOR TRANSFER OF DATA
C TO TAPE.

380     OPEN(UNIT=1,DEVICE='DSK',ACCESS='SEQOUT',
        1DISPOSE='SAVE',FILE='SOL.DAT')
        N5=1
        BGSOL=1
        LPSOL=1
        CALL SOL
        CLOSE(UNIT=1)
        N5=5


445     OPEN(UNIT=24,DEVICE='DSK',ACCESS='SEQOUT',
        1dISPOSE='SAVE',FILE='YBUSM.DAT')
        WRITE(24,3030)
3030    FORMAT(15X,'**YBUS MATRIX**')
        DO 450 I=1,MBUS
        WRITE(24,3040)I
3040    FORMAT(1X,'ROW',I3)
450     WRITE(24,3050) (YMAG(I,J),YANG(I,J),J=1,MBUS)
3050    FORMAT(6E12.5)

        IF(KINV.EQ.0)GO TO 465
        WRITE(24,3060)
3060    FORMAT(15X,'**ZBUS MATRIX**')
        DO 460 I=1,MBUS
        WRITE(24,3040)I
460     WRITE(24,3070) (ZMAG(I,J),ZANG(I,J),J=1,MBUS)
3070    FORMAT(6E12.5)
465     CLOSE(UNIT=24)
        GO TO 330
500     CONTINUE
        STOP
        END


        SUBROUTINE LDATA
C SUBROUTINE TO READ, WRITE, AND CHANGE LINE DATA
```

```
        REAL LNGTH
        INTEGER ABORT,BUSM,CL,CHL,EB,SB,TLDAT
        DOUBLE PRECISION AJ1,AJ2,AJ3,AJ4,BJ1,BJ2,BJ3,BJ4,
        1CALP,CALQ,DQ,DVANG,DVMAG,G,H,ITL,P,Q,VANG,VMAG,
        2XX,YANG,YMAG,YY,YYSHT,ZANG,ZMAG,YSER
        COMMON/ /          AJ1(15,15),    AJ2(15,15),     AJ3(15,15),
        1AJ4(15,15),       ALPHA(15),     BDAT,           BETA(15),
        1BGSOL,            BJ1(15,15),    BJ2(15,15),     BJ3(15,15),
        1BJ4(15,15),       BTYPE(15),     BUSM,           CALP(15),
        1CALQ(15),         CB,CC,           CG,             CHB,
        1CHG,              CHL,           CL,             CHY,
        1CHZ,              DQ(15),        DVANG(15),      DVMAG(15),
        1EB(30),           ESP1,          ESP2,           G(15),
        1GAMMA(15),        GDAT,          H(15),          IC(15),
        1II,               ITER,          ITL(15),        KINV,
        1KSOL,             LAMBDA,        LDAT,           LITER,
        1LNGTH(30),        LNO,           LOSS,           LPSOL,
        1LSET,             MBUS,          NS,             NSWGB,
        1P(15),            PD(15),        PG(15),         PGMAX(15),
        1PGMIN(15),        Q(15),         QD(15),QDAT,        QG(15),
        1QGMAX(15),        QGMIN(15),     SB(30),         YYSHT(30,2),
        1TOL,TLDAT,        TPD,TBDAT,     TPL,TGDAT,      VANG(15),
        1VMAG(15),         VSPEC(15),     XX(15,15),      YANG(15,15),
        1YMAG(15,15),      YSHT(30,2),    YY(15,15),      ZANG(15,15),
        1ZMAG(15,15),      ZSER(30,2),    YSER(30,2),     ABORT

        ILD=0
        IF(LDAT.EQ.1)GO TO 50
        IF(TLDAT.EQ.'Y')GO TO 20

C READ LINE DATA FOR SOS FILE FOR20.DAT
        OPEN(UNIT=20,DEVICE='DSK',ACCESS='SEQIN',DISPOSE='SAVE',
        1FILE='FOR20.DAT')
10      DO 5 I=1,LNO
5       READ(20,*) K,SB(I),EB(I),LNGTH(I),YSHT(I,1),YSHT(I,2),
        1ZSER(I,1),ZSER(I,2)
        CLOSE(UNIT=20)
2000    FORMAT(3I,5F)
        IF(K.EQ.LNO)GO TO 40
        TYPE 8
8       FORMAT(1X,'TOTAL NUMBER OF LINES ENTERED DOES NOT EQUAL
        1NUMBER OF DATA LINES IN FOR20.DAT')
        ABORT=1
        LDAT=0
        RETURN

C READ LINE DATA FROM CRT/TTY
20      OPEN(UNIT=20,DEVICE='DSK',ACCESS='SEQOUT',DISPOSE='SAVE',
        1FILE='FOR20.DAT')
        IF(ILD.EQ.1)GO TO 21
        TYPE 1000
1000    FORMAT(/1X,'**SYSTEM LINE DATA**'/1X,'>>ENTER (LINE #,
        1SB #,EB #, LINE LENGTH, YSHUNT G,YSHUNT B,
        2ZSERIES R,ZSERIES X)')
21      DO 30 I=1,LNO
        IF(ILD.EQ.1)GO TO 25
22      ACCEPT 2002,K,SB(I),EB(I),LNGTH(I),YSHT(I,1),YSHT(I,2),
        1ZSER(I,1),ZSER(I,2)
2002    FORMAT(3I,5F)
```

```
         IF(K.EQ.I)GO TO 25
         TYPE 1002,I
1002     FORMAT(1X,'WRONG ENTRY, LINE # SHOULD=',I3,', TRY AGAIN')
         GO TO 22
25       WRITE(20,1004)I,SB(I),EB(I),LNGTH(I),YSHT(I,1),YSHT(I,2),
         1ZSER(I,1),ZSER(I,2)
1004     FORMAT(3I3,5E12.5)
30       CONTINUE
         CLOSE(UNIT=20)
         IF(ILD.EQ.1)GO TO 180
40       BUSM=0
         LDAT=1

C CHECK FOR LINE DATA PRINT OUT
50       IF(CL.EQ.'N')GO TO 70
         TYPE 1010
1010     FORMAT(/1X,'**SYSTEM LINE DATA**'/1X,'LINE #',1X,
         1'SB',1X,'EB',4X,'LENGTH',12X,'Y SHUNT',19X,'Z SERIES')
         DO 60 I=1,LNO
60       TYPE 1020,I,SB(I),EB(I),LNGTH(I),YSHT(I,1),YSHT(I,2),
         1ZSER(I,1),ZSER(I,2)
1020     FORMAT(1X,I3,2X,2I3,1X,E12.5,2(1X,2E12.5))
         CL='N'

         IF(ILD.EQ.1)GO TO 80
C CHECK FOR LINE DATA CHANGES
70       IF(CHL.EQ.'N')GO TO 180
         TYPE 1030
1030     FORMAT(/1X,'** LINE DATA CHANGES**'/
         11x,'>>ENTER (VARIABLE #, LINE #, NEW VALUE THEN "CR")'/
         110X,'1=STARTING BUS #'/10X,'2=ENDING BUS #'/10X,'3=LINE
         1LENGTH'/10X,'4=Y SHUNT G'/10X,'5=Y SHUNT B'/10X,'6=Z SERIES R'/
         210X,'7=Z SERIES X'/10X,'8=RETYPE COPY OF ALL LINE DATA'/10X,'
         19=CHANGE TOTAL NUMBER OF LINES'/10X,'>>ENTER 0 THEN
         1"CR" WHEN CHANGES COMPLETED')
80       READ(5,2010)NAME,I,X
2010     FORMAT(2I,F)
         IF(NAME.EQ.0)GO TO 175
         IF(NAME.EQ.8)GO TO 172
         IF(NAME.EQ.9)GO TO 173
         IF(((NAME.GE.1).AND.(NAME.LE.7)).AND.((I.GE.0).AND.(I.LE.LNO)))
         1GO TO 100
         TYPE 1040,LNO
1040     FORMAT(1X,'WRONG ENTRY, VARIABLE # MUST BE 1-7,'/
         11x,'LINE # MUST BE 1 TO',I3,', TRY AGAIN')
         GO TO 80
100      BUSM=0
         GO TO(110,120,130,140,150,160,170)NAME
110      SB(I)=IFIX(X)
         TYPE 1060, I,SB(I)
1060     FORMAT(1X,'SB(LINE #',I3,')=',I3)
         GO TO 80
120      EB(I)=IFIX(X)
         TYPE 1070,I,EB(I)
1070     FORMAT(1X,'EB(LINE #',I3,')=',I3)
         GO TO 80
130      LNGTH(I)=X
         TYPE 1080,I,LNGTH(I)
1080     FORMAT(1X,'LENGTH(LINE #',I3,')=',E12.5)
         GO TO 80
140      YSHT(I,1)=X
```

```
       TYPE 1090,I,YSHT(I,1)
1090   FORMAT(1X,'YSHT G (LINE #',I3,')=',E12.5)
       GO TO 80
150    YSHT(I,2)=X
       TYPE 1100,I,YSHT(I,2)
1100   FORMAT(1X,'YSHT B (LINE #',I3,')=',E12.5)
       GO TO 80
160    ZSER(I,1)=X
       TYPE 1110,I,ZSER(I,1)
1110   FORMAT(1X,'ZSER R (LINE #',I3,')=',E12.5)
       GO TO 80
170    ZSER(I,2)=X
       TYPE 1120,I,ZSER(I,2)
1120   FORMAT(1X,'ZSER X (LINE #',I3,')=',E12.5)
       GO TO 80
172    ILD=1
       CL='Y'
       GO TO 50
173    LNO=I
       BUSM=0
       TYPE 1130,I
1130   FORMAT(1X,'TOTAL NUMBER OF LINES =',I3)
       GO TO 80
175    CHL='N'
       ILD=1
       GO TO 20
180    ILD=0
       RETURN
       END


       SUBROUTINE BDATA
C SUBROUTINE TO READ,WRITE,AND CHANGE BUS DATA
       INTEGER ABORT,BDAT,BTYPE,CB,CHB,MBUS,TBDAT
       DOUBLE PRECISION AJ1,AJ2,AJ3,AJ4,BJ1,BJ2,BJ3,BJ4,
       1CALP,CALQ,DQ,DVANG,DVMAG,G,H,ITL,P,Q,VANG,VMAG,
       2XX,YANG,YMAG,YY,YYSHT,ZANG,ZMAG,YSER
       COMMON/ /       AJ1(15,15),   AJ2(15,15),   AJ3(15,15),
       1AJ4(15,15),    ALPHA(15),    BDAT,         BETA(15),
       1BGSOL,         BJ1(15,15),   BJ2(15,15),   BJ3(15,15),
       1BJ4(15,15),    BTYPE(15),    BUSM,         CALP(15),
       1CALQ(15),      CB,CC,          CG,           CHB,
       1CHG,           CHL,          CL,           CMY,
       1CMZ,           DQ(15),       DVANG(15),    DVMAG(15),
       1EB(30),        ESP1,         ESP2,         G(15),
       1GAMMA(15),     GDAT,         H(15),        IC(15),
       1II,            ITER,         ITL(15),      KINV,
       1KSOL,          LAMBDA,       LDAT,         LITER,
       1LNGTH(30),     LNO,          LOSS,         LPSOL,
       1LSET,          MBUS,         NS,           NSWGB,
       1P(15),         PD(15),       PG(15),       PGMAX(15),
       1PGMIN(15),     Q(15),        QD(15),QDAT,      QG(15),
       1QGMAX(15),     QGMIN(15),    SB(30),       YYSHT(30,2),
       1TOL,TLDAT,     TPD,TBDAT,    TPL,TGDAT,    VANG(15),
       1VMAG(15),      VSPEC(15),    XX(15,15),    YANG(15,15),
       1YMAG(15,15),   YSHT(30,2),   YY(15,15),    ZANG(15,15),
       1ZMAG(15,15),   ZSER(30,2),   YSER(30,2),   ABORT

       IBD=0
       IF(BDAT.EQ.1)GO TO 50
```

```
        IF(TBDAT.EQ.'Y')GO TO 20

C READ BUS DATA FROM SOS FILE FOR21.DAT
        OPEN(UNIT=21,DEVICE='DSK',ACCESS='SEQIN',DISPOSE='SAVE',
        1FILE='FOR21.DAT')
10      DO 15 I=1,NBUS
15      READ(21,*) K,BTYPE(I),PG(I),QG(I),PD(I),QD(I),VSPEC(I)
2000    FORMAT(2I,5F)
        CLOSE(UNIT=21)
        IF(K.EQ.NBUS)GO TO 45
        TYPE 500
500     FORMAT(1X,'TOTAL NUMBER OF BUSES ENTERED DOES NOT
        1EQUAL NUMBER OF DATA LINES IN FOR21.DAT')
        BDAT=0
        ABORT=1
        RETURN



C READ BUS DATA FROM CRT/TTY
19      OPEN(UNIT=21,DEVICE='DSK',ACCESS='SEQOUT',DISPOSE='SAVE',
        1FILE='FOR21.DAT')
        IF(IBD.EQ.1)GO TO 21
20      TYPE1000
1000    FORMAT(/1X,'**SYSTEM BUS DATA**'/1X,'>>ENTER(BUS#,
        1 BUS TYPE,PG,QG,PD,QD,VSPEC)')
21      DO 30 I=1,NBUS
        IF(IBD.EQ.1)GO TO 25
22      ACCEPT 2000,K,BTYPE(I),PG(I),QG(I),PD(I),QD(I),VSPEC(I)
        IF(K.EQ.I)GO TO 25
        TYPE 1005,I
1005    FORMAT(1X,'WRONG ENTERY, BUS # SHOULD=',I3,',TRY AGAIN.')
        GO TO 22
25      WRITE(21,1010)I,BTYPE(I),PG(I),QG(I),PD(I),QD(I),VSPEC(I)
1010    FORMAT(2I3,5F12.5)
30      CONTINUE
        CLOSE(UNIT=21)
45      BDAT=1
        IF(IBD.EQ.1)GO TO 170

C CHECK FOR PRINTOUT OF BUS DATA
50      IF(CB.EQ.'N')GO TO 70
        TYPE 1020
1020    FORMAT(/1X,'**SYSTEM BUS DATA**'/2X,'BUS',3X,'TYPE',
        15X,'PG',8X,'QG',8X,'PD',8X,'QD',7X,'VSPEC')
        DO 60 I=1,NBUS
        TYPE 1030,I,BTYPE(I),PG(I),QG(I),PD(I),QD(I),VSPEC(I)
1030    FORMAT(1X,I3,4X,I3,5F10.5)
60      CONTINUE
        CB='N'
        IF(IBD.EQ.1)GO TO 80

C CHECK FOR BUS DATA CHANGES
70      IF(CHB.EQ.'N')GO TO 170
        TYPE 1040
1040    FORMAT(/1X,'**BUS DATA CHANGES**'/
        11X,'>>ENTER(VARIABLE #, BUS #, NEW VALUE THEN "CR"'/
        110X,'1=BUS TYPE'/10X,'2=PG'/10X,'3=QG'/10X,'4=PD'
        1/10X,'5=QD'/10X,'6=VSPEC'/10X,'7=RETYPE COPY OF ALL BUS DATA'
        1/10X,'8=CHANGE TOTAL NUMBER OF BUSES'/10X,
        1'>>ENTER 0 THEN "CR" WHEN CHANGES COMPLETED.')
80      READ(5,2010)NAME,I,X
2010    FORMAT(2I,F)
        IF(NAME.EQ.0)GO TO 165
        IF(NAME.EQ.8)GO TO 163
        IF(((NAME.GE.1).AND.(NAME.LE.7)).AND.
        1((I.GE.0).AND.(I.LE.MBUS)))GO TO 100
        TYPE 1050,MBUS
1050    FORMAT(1X,'WRONG ENTRY, VARIABLE NAME MUST BE 1-7,'/
        11X,'BUS # MUST BE 1 TO',I3,',TRY AGAIN.')
        GO TO 80
100     GO TO(110,120,130,140,150,160,162) NAME
110     BTYPE(I)=IFIX(X)
        TYPE 1060,I,BTYPE(I)
1060    FORMAT(1X,'BUS TYPE(BUS #',I3,')=',I3)
        GO TO 80
120     PG(I)=X
        TYPE 1070,I,PG(I)
1070    FORMAT(1X,'PG(BUS #',I3,')=',E12.5)
        GO TO 80
130         QG(I)=X
        TYPE 1080,I,QG(I)
1080    FORMAT(1X,'QG(BUS #',I3,')=',E12.5)
        GO TO 80
140     PD(I)=X
        TYPE 1090,I,PD(I)
1090    FORMAT(1X,'PD(BUS #',I3,')=',E12.5)
        GO TO 80
150     QD(I)=X
        TYPE 1100,I,QD(I)
1100    FORMAT(1X,'QD(BUS #',I3,')=',E12.5)
        GO TO 80
160     VSPEC(I)=X
        TYPE 1110,I,VSPEC(I)
1110    FORMAT(1X,'VSPEC(BUS #',I3,')=',E12.5)
        GO TO 80
162     CB='Y'
        IBD=1
        GO TO 50
163     MBUS=I
        TYPE 1120,I
1120    FORMAT(1X,'TOTAL NUMBER OF BUSES =',I3)
        GO TO 80
165     CHB='N'
        IBD=1
        GO TO 19

170     IBD=0
        RETURN
        END


        SUBROUTINE GDATA
C SUBROUTINE TO READ,WRITE,AND CHANGE GENERATOR DATA

        INTEGER BTYPE,CG,CHG,GDAT,GNO,MBUS,TGDAT,QDAT
        DOUBLE PRECISION AJ1,AJ2,AJ3,AJ4,BJ1,BJ2,BJ3,BJ4,
        1CALP,CALQ,DC,DVANG,DVMAG,G,R,ITL,P,Q,VANG,VMAG,
        2XX,YANG,YMAG,YY,YYSHT,ZANG,ZMAG,YSER
        COMMON/ /      AJ1(15,15),    AJ2(15,15),    AJ3(15,15),
        1AJ4(15,15),    ALPHA(15),     BDAT,          BETA(15),
        1BGSOL,         BJ1(15,15),    BJ2(15,15),    BJ3(15,15),
        1BJ4(15,15),    BTYPE(15),     BUSN,          CALP(15),
        1CALQ(15),      CB,CC,           CG,            CHB,
        1CHG,           CHL,           CL,            CHY,
        1CHZ,           DQ(15),        DVANG(15),     DVMAG(15),
        1EB(30),        ESP1,          ESP2,          G(15),
        1GAMMA(15),     GDAT,          H(15),         IC(15),
        1II,            ITER,          ITL(15),       KINV,
        1KSOL,          LAMBDA,        LDAT,          LITER,
        1LNGTH(30),     LNO,           LOSS,          LPSOL,
        1LSET,          NBUS,          N5,            NSWGB,
        1P(15),         PD(15),        PG(15),        PGMAX(15),
        1PGMIN(15),     Q(15),         QD(15),QDAT,        QG(15),
        1QGMAX(15),     QGMIN(15),     SB(30),        YYSHT(30,2),
        1TOL,TLDAT,     TPD,TBDAT,     TPL,TGDAT,     VANG(15),
        1VMAG(15),      VSPEC(15),     XX(15,15),     YANG(15,15),
        1YMAG(15,15),   YSHT(30,2),    YY(15,15),     ZANG(15,15),
        1ZMAG(15,15),   ZSER(30,2),    YSER(30,2),    ABORT

        IGB=0
        IF(GDAT.EQ.1)GO TO 56
        IF(KINV.EQ.0)GO TO 25
        IF(TGDAT.EQ.'Y')GO TO 20

C READ GENERATOR DATA FROM SOS FILE FOR22.DAT
        OPEN(UNIT=22,DEVICE='DSK',ACCESS='SEQIN',DISPOSE='SAVE',
        1FILE='FOR22.DAT')
        GNO=0
10      DO 15 I=1,NBUS
        IF((BTYPE(I)).GE.0)GO TO 15
        READ(22,*)K,ALPHA(I),BETA(I),GAMMA(I)
2000    FORMAT(I,3F)
        GNO=GNO+1
15      CONTINUE
        CLOSE(UNIT=22)
        GO TO 40

C READ GENERATOR DATA FROM CRT/TTY
20      OPEN(UNIT=22,DEVICE='DSK',ACCESS='SEQOUT',DISPOSE='SAVE',
        1FILE='FOR22.DAT')
        IF(IGB.EQ.1)GO TO 29
        TYPE 1000
1000    FORMAT(/1X,'**GENERATOR DATA**'/1X,'>>ENTER(BUS#,ALPHA,
        1BETA,GAMMA)')
29      DO 30 I=1,NBUS
        IF(BTYPE(I).GE.0)GO TO 30
        IF(IGB.EQ.1)GO TO 33
32      ACCEPT 2002,K,ALPHA(I),BETA(I),GAMMA(I)
2002    FORMAT(I,3F)
        IF(K.EQ.I)GO TO 31
        TYPE 1010,I
1010    FORMAT(1X,'WRONG ENTRY, BUS # SHOULD=',I3,', TRY AGAIN.')
        GO TO 32
31      GNO=GNO+1
33      WRITE(22,2003)I,ALPHA(I),BETA(I),GAMMA(I)
2003    FORMAT(I3,3F)
30      CONTINUE
        CLOSE(UNIT=22)
        IF(IGB.EQ.1)GO TO 48
40      GDAT=1
```

```
C INITIALIZING GENERATOR AND Q CONSTRAINTS
25      IF(QDAT.EQ.1)GO TO 56
        IF(TGDAT.EQ.'N')GO TO 50

C READ CONSTRAINT DATA FROM TTY/CRT
        TYPE 1006
1006    FORMAT(1X,'**GENERATOR CONSTRAINT DATA**'/1X,'>>ENTER FOR
        1TYPE  -1 & -3  BUS (BUS #,PGMAX,PGMIN)'/9X,'FOR TYPE  -2 BUS
        1        (BUS #,PGMAX,PGMIN,QGMAX,QGMIN)'/9X,'FOR TYPE  +2 BUS
        1       (BUS #,QGMAX,QGMIN)')
        DO 36 I=1,MBUS
35      IF(BTYPE(I).GE.0)GO TO 38
        IF(BTYPE(I).EQ.-2)GO TO 37
        ACCEPT 2005,K,PGMAX(I),PGMIN(I)
2005    FORMAT(I,2F)
        GO TO 39
37      ACCEPT 2020,K,PGMAX(I),PGMIN(I),QGMAX(I),QGMIN(I)
2020    FORMAT(I,4F)
        GO TO 39
38      IF(BTYPE(I).NE.2)GO TO 36
        ACCEPT 2005,K,QGMAX(I),QGMIN(I)
39      IF(K.EQ.I)GO TO 36
        TYPE 1015,I
1015    FORMAT(1X,'WRONG ENTRY,BUS # SHOULD BE=',I3,',TRY AGAIN.')
        GO TO 35
36      CONTINUE

C ENTER CONSTRAINT DATA INTO SOS FILE FOR23.DAT
48      OPEN(UNIT=23,DEVICE='DSK',ACCESS='SEQOUT',DISPOSE='SAVE',
        1FILE='FOR23.DAT')

        DO 52 I=1,MBUS
        IF(BTYPE(I).GE.0)GO TO 53
        IF (BTYPE(I).EQ.-2)GO TO 51
        WRITE(23,2007)I,PGMAX(I),PGMIN(I)
2007    FORMAT(1X,I3,2F11.4)
        GC TO 52
51      WRITE(23,2006)I,PGMAX(I),PGMIN(I),QGMAX(I),QGMIN(I)
2006    FORMAT(1X,I3,4F11.4)
        GO TO 52
53      IF (BTYPE(I).NE.2)GO TO 52
        WRITE(23,2009)I,QGMAX(I),QGMIN(I)
2009    FORMAT(1X,I3,22X,2F11.4)
52      CONTINUE
        CLOSE(UNIT=23)
        IF(IGB.EQ.1)GO TO 180
        GO TO 54
C READ CONSTRAINT DATA FROM SOS FILE FOR23.DAT
50      OPEN(UNIT=23,DEVICE='DSK',ACCESS='SEQIN',DISPOSE='SAVE',
        1FILE='FOR23.DAT')
        DO 46 I=1,MBUS
        IF(BTYPE(I).GE.0)GO TO 42
        IF(BTYPE(I).EQ.-2)GO TO 41
        READ(23,*)K,PGMAX(I),PGMIN(I)
        GO TO 46
41      READ(23,2020)K,PGMAX(I),PGMIN(I),QGMAX(I),QGMIN(I)
        GO TO 46
42      IF(BTYPE(I).NE.2)GO TO 46
        READ(23,2005)K,QGMAX(I),QGMIN(I)
```

```
.46       CONTINUE
          CLOSE(UNIT=23)
54        QDAT=1

C CHECK FOR PRINTOUT OF GENERATOR DATA
56        IF(CG.EQ.'N')GO TO 70
          IF(KINV.EQ.0)GO TO 61
          TYPE 1020,GNO
1020      FORMAT(/1X,'**SYSTEM GENERATOR DATA**'/1X,'NUMBER
          1OF GENERATORS =',I3/1X,'BUS#',4X,
          1'ALPHA',4X,'BETA',6X,'GAMMA')
          DO 60 I=1,MBUS
          IF((BTYPE(I).GE.0))GO TO 60
          TYPE 1030,I,ALPHA(I),BETA(I),GAMMA(I)
1030      FORMAT(1X,I3,3F10.5)
60        CONTINUE

C         PRINT GENERATOR CONSTRAINTS
61        WRITE(5,1032)
1032      FORMAT(/1X,'**GENERATOR CONSTRAINTS**'/1X,'BUS #',
          13X,'PGMAX',5X,'PGMIN',7X,'QGMAX',6X,'QGMIN')
          DO 65 I=1,MBUS
          IF((BTYPE(I)).GE.0)GO TO 64
          IF(BTYPE(I).EQ.-2)GO TO 62
          WRITE(5,1034)I,PGMAX(I),PGMIN(I)
1034      FORMAT(1X,I3,2F11.4)
          GO TO 65
62        WRITE(5,1036)I,PGMAX(I),PGMIN(I),QGMAX(I),QGMIN(I)
1036      FORMAT(1X,I3,4F11.4)
          GO TO 65
64        IF(BTYPE(I).NE.2)GO TO 65
          WRITE(5,1038)I,QGMAX(I),QGMIN(I)
1038      FORMAT(1X,I3,22X,2F11.4)
65        CONTINUE
66        CG='N'
          IF(IGB.EQ.1)GO TO 80

C CHECK FOR GENERATOR DATA CHANGES
70        IF(CHG.EQ.'N')GO TO 180
          TYPE 1040
1040      FORMAT(/1X,'**GENERATCR DATA CHANGES**'/
          11X,'>>ENTER (VARIABLE #, BUS #, NEW VALUE THEN "CR")'/
          110X,'1=ALPHA'/10X,'2=BETA'/10X,'3=GAMMA'/10X,'4=PGMAX'/10X,'
          25=PGMIN'/10X,'6=QGMAX'/10X,'7=QGMIN'/10X,'8=RETYPE ALL
          1GENERATOR DATA'/10X,'>>ENTER 0 THEN "CR"
          3WHEN CHANGES COMPLETED.')
80        READ(5,2010)NAME,I,X
2010      FORMAT(2I,F)
          IF(NAME.EQ.0)GO TO 175
          IF(((NAME.GE.1).AND.(NAME.LE.8)).AND.((I.GE.0).AND.
          1(I.LE.MBUS)))
          2GO TO 90
          TYPE 1050,MBUS
1050      FORMAT(1X,'WRONG ENTRY, VARIABLE MUST BE 1-8,'/
          11X,'BUS # MUST BE 0 TO',I3,',CHECK BUS TYPE IS'/
          11X,'NEGATIVE,AND TRY AGAIN.')
          GO TO 80
90        CONTINUE
100       GO TO(110,120,130,140,150,160,170,172)NAME
110       ALPHA(I)=X
          TYPE 1060,I,ALPHA(I)
```

```
1060      FORMAT(1X,'ALPHA(BUS #',I3,')=',E12.5)
          GO TO 80
120       BETA(I)=X
          TYPE 1070,I,BETA(I)
1070      FORMAT(1X,'BETA(BUS #',I3,')='E12.5)
          GO TO 80
130       GAMMA(I)=X
          TYPE 1080,I,GAMMA(I)
1080      FORMAT(1X,'GAMMA(BUS #',I3,')=',E12.5)
          GO TO 80
140       PGMAX(I)=X
          TYPE 1090,I,PGMAX(I)
1090      FORMAT(1X,'PGMAX(BUS #',I3,')=',E12.5)
          GO TO 80
150       PGMIN(I)=X
          TYPE 1100,I,PGMIN(I)
1100      FORMAT(1X,'PGMIN(BUS #',I3,')=',E12.5)
          GO TO 80
160       QGMAX(I)=X
          TYPE 1110,I,QGMAX(I)
1110      FORMAT(1X,'QGMAX(BUS #',I3,')=',E12.5)
          GO TO 80
170       QGMIN(I)=X
          TYPE 1120,I,QGMIN(I)
1120      FORMAT(1X,'QGMIN(BUS #',I3,')=',E12.5)

          GO TO 80
172       CG='Y'
          IGB=1
          GO TO 56

175       CHG='N'
          IGB=1
          IF(KINV.EQ.0)GO TO 48
          GO TO 20


180       IGB=0
          RETURN
          END


          SUBROUTINE YBUSM
C SUBROUTINE TO CALCULATE YBUS AND ZBUS MATRIX

          REAL LNGTH
          INTEGER BUSM,CHY,CHZ,EB,EBUS,SB
          DOUBLE PRECISION AJ1,AJ2,AJ3,AJ4,BJ1,BJ2,BJ3,BJ4,
          1CALP,CALQ,DQ,DVANG,DVMAG,G,H,ITL,P,Q,VANG,VMAG,
          2XX,YANG,YMAG,YY,YYSHT,ZANG,ZMAG,YSER,D

          COMMON/ /       AJ1(15,15),    AJ2(15,15),    AJ3(15,15),
          1AJ4(15,15),    ALPHA(15),     BDAT,          BETA(15),
          1BGSOL,         BJ1(15,15),    BJ2(15,15),    BJ3(15,15),
          1BJ4(15,15),    BTYPE(15),     BUSM,          CALP(15),
          1CALQ(15),      CB,CC,            CG,            CHB,
          1CHG,           CHL,           CL,            CHY,
          1CHZ,           DQ(15),        DVANG(15),     DVMAG(15),
          1EB(30),        ESP1,          ESP2,          G(15),
          1GAMMA(15),     GDAT,          H(15),         IC(15),
        1II,           ITER,          ITL(15),       KINV,
        1KSOL,         LAMBDA,        LDAT,          LITER,
        1LNGTH(30),    LNO,           LOSS,          LPSOL,
        1LSET,         MBUS,          NS,            NSNGB,
        1P(15),        PD(15),        PG(15),        PGMAX(15),
        1PGMIN(15),    Q(15),         QD(15),QDAT,       QG(15),
        1QGMAX(15),    QGMIN(15),     SB(30),        YYSHT(30,2),
        1TOL,TLDAT,    TPD,TBDAT,     TPL,TGDAT,     VANG(15),
        1VMAG(15),     VSPEC(15),     XX(15,15),     YANG(15,15),
        1YMAG(15,15),  YSHT(30,2),    YY(15,15),     ZANG(15,15),
        1ZMAG(15,15),  ZSEB(30,2),    YSER(30,2),    ABOPT

        IF((BUSM:EQ.2).OR.((KINV.EQ.0).AND.(BUSM.EQ.1)))GO TO 90
        TYPE 1000

1000    FORMAT(/1X,'**GENERATING YBUS MATBIX**')

C CALCULATING LINE SHONT AND SERIES ADMITTANCE
        DO 15 I=1,LNO
        YYSHT(I,1)=(DBLE(YSHT(I,1))*DBLE(LNGTH(I)))/2.0D0
        YYSHT(I,2)=(DBLE(YSHT(I,2))*DBLE(LNGTH(I)))/2.0D0
        D=(DBLE(LNGTH(I)))*((DBLE(ZSER(I,1))*DBLE(ZSER(I,1)))
        1+(DBLE(ZSER(I,2))*DBLE(ZSER(I,2))))
        IF(D.EQ.0.0D0)GO TO 20
        YSER(I,1)=DBLE(ZSER(I,1))/D
        YSER(I,2)=-DBLE(ZSER(I,2))/D
        GO TO 15
20      YSER(I,1)=0.0D0
        YSER(I,2)=0.0D0
15      continue

C CALCULATING YBUS MATRIX
        DO 30 I=1,MBUS
        DO 30 J=1,MBUS
        YMAG(I,J)=0.0D0
        YANG(I,J)=0.0D0
        AJ1(I,J)=0.0D0
        AJ2(I,J)=0.0D0
        IF(J.EQ.I)GO TO 60

C OFF DIAGONAL ELEMENTS(I.NE.J)
        DO 40 L=1,LNO
        IF(((SB(L).EQ.I).AND.(EB(L).EQ.J)).OR.((SB(L).EQ.J)
        1.AND.(EB(L).EQ.I)))GO TO 42
        GC TO 40
42      YMAG(I,J)=DSQRT((YSER(L,1)*YSER(L,1))+(YSER(L,2)*YSER(L,2)))
        IF(YSER(L,1).EQ.(0.0D0))GO TO 22
        YANG(I,J)=DATAN2(-YSER(L,2),-YSEB(L,1))
        GO TO 29
22      IF(YSER(L,2))24,26,28
24      YANG(I,J)=1.570796326794897
        GO TO 29
26      YANG(I,J)=0.0D0
        GO TO 29
28      YANG(I,J)=-1.570796326794897
29      IF(KINV.EQ.0)GO TO 40
        AJ1(I,J)=-YSER(L,1)
        AJ2(I,J)=-YSER(L,2)
40      CONTINUE
        GO TO 30
```

```
C ON DIAGONAL ELEMENTS(I.EQ.J)
60      DO 45 L=1,LNO
        IF((SB(L).EQ.I).OR.(EB(L).EQ.I))GO TO 46
        GO TO 45
46      AJ1(I,J)=AJ1(I,J)+YYSHT(L,1)+YSER(L,1)
        AJ2(I,J)=AJ2(I,J)+YYSHT(L,2)+YSER(L,2)
45      CONTINUE
        YMAG(I,J)=DSQRT(AJ1(I,J)*AJ1(I,J)+AJ2(I,J)*AJ2(I,J))
        IF(AJ1(I,J).EQ.(0.0D0))GO TO 31
        YANG(I,J)=DATAN2(AJ2(I,J),AJ1(I,J))
        GO TO 30
31      IF(AJ2(I,J))32,33,34
32      YANG(I,J)=-1.570796326794897
        GO TO 30
33      YANG(I,J)=0.0D0
        GO TO 30
34      YANG(I,J)=+1.570796326794897
30      CONTINUE
35      BUSM=1
        IF(KINV.EQ.0)GO TO 90

C CALCULATING ZBUS MATRIX
        TYPE 1005
1005    FORMAT(15X,'**CALCULATING ZBUS MATRIX**')
        DO 50 I=1,MBUS
        DO 50 J=1,MBUS
        BJ1(I,J)=AJ1(I,J)
50      BJ2(I,J)=AJ2(I,J)
        BJ3(I,J)=0.0D0
        BJ4(I,J)=0.0D0
        CALL CTRI(BJ1,BJ2,BJ3,BJ4,MBUS)
        CALL CTRIV(BJ3,BJ4,BJ1,BJ2,MBUS)
        DO 80 I=1,MBUS
        DO 80 J=1,MBUS
        ZMAG(I,J)=DSQRT(BJ1(I,J)*BJ1(I,J)+BJ2(I,J)*BJ2(I,J))
        IF(BJ1(I,J).EQ.0.0D0)GO TO 82
80      ZANG(I,J)=DATAN2(BJ2(I,J),BJ1(I,J))
        GO TO 81
82      IF(BJ2(I,J))83,84,85
83      ZANG(I,J)=-1.570796326794897
        GO TO 81
84      ZANG(I,J)=0.0D0
        GO TO 81
85      ZANG(I,J)=1.570796326794897
81      BUSM=2/

C PRINTOUT OF YBUS AND ZBUS MATRIX
90      IF(CMY.EQ.'N')GO TO 105
        TYPE 4000
4000    FORMAT(/1X,'**CALCULATED SYSTEM LINE DATA**'/1X,
        1'LINE #',15X,'Y SHUNT',23X,'Y SERIES')
        DO 95 I=1,LNO
        TYPE 4100,I,YYSHT(I,1),YYSHT(I,2),YSER(I,1),YSER(I,2)
4100    FORMAT(I4,4X,2E14.6,2X,2E14.6)
95      CONTINUE
        TYPE 1010
1010    FORMAT(/1X,'**YBUS MATRIX**')
        DO 100 I=1,MBUS
        TYPE 1020,I
1020    FORMAT(1X,'ROW',I3)
```

```
        TYPE 1030,(YMAG(I,J),YANG(I,J),J=1,NBUS)
1030    FORMAT(6E12.5)
100     CONTINUE
        CHY='N'
105     IF(KINV.EQ.0)GO TO 120
        IF(CHZ.EQ.'N')GO TO 120
        TYPE 1040
1040    FORMAT(/1X,'**ZBUS MATRIX**')
        DO 110 I=1,NBUS
        TYPE 1020,I
        TYPE 1030,(ZMAG(I,J),ZANG(I,J),J=1,NBUS)
110     CONTINUE
        CHZ='N'

120     RETURN
        END


        SUBBOUTINE NBLF
C        CALCULATES P,Q,V,AND ANGLE AT EACH BUS
        INTEGER ABORT,BTYPE, NBUS
        DIMENSION DP(15)

        DOUBLE PRECISION AJ1,AJ2,AJ3,AJ4,BJ1,EJ2,BJ3,BJ4,
        1CALP,CALQ,DQ,DVANG,DVMAG,G,H,ITL,P,C,VANG,VMAG,
        2XX,YANG,YMAG,YY,YYSHT,ZANG,ZMAG,YSER,GMAX,HMAX,DP

        COMMON/ /       AJ1(15,15),     AJ2(15,15),     AJ3(15,15),
        1AJ4(15,15),    ALPHA(15),      BDAT,           BETA(15),
        1BGSOL,         BJ1(15,15),     BJ2(15,15),     BJ3(15,15),
        1BJ4(15,15),    BTYPE(15),      BUSH,           CALP(15),
        1CALQ(15),      CB,CC,             CG,             CHB,
        1CHG,           CHL,            CL,             CHY,
        1CHZ,           DQ(15),         DVANG(15),      DVMAG(15),
        1EB(30),        ESP1,           ESP2,           G(15),
        1GAMMA(15),     GDAT,           H(15),          IC(15),
        1II,            ITER,           ITL(15),        KINV,
        1KSOL,          LAMBDA,         LDAT,           LITER,
        1LNGTH(30),     LNO,            LOSS,           LPSOL,
        1LSET,          NBUS,           NS,             NSWGB,
        1P(15),         PD(15),         PG(15),         PGMAX(15),
        1PGMIN(15),     Q(15),          QD(15),QDAT,       QG(15),
        1QGMAX(15),     QGMIN(15),      SB(30),         YYSHT(30,2),
        1TOL,TLDAT,     TPD,TBDAT,      TPL,TGDAT,      VANG(15),
        1VMAG(15),      VSPEC(15),      XX(15,15),      YANG(15,15),
        1YMAG(15,15),   YSHT(30,2),     YY(15,15),      ZANG(15,15),
        1ZMAG(15,15),   ZSER(30,2),     YSER(30,2),     ABORT,IBT(15)

        LL=0
        TYPE 100
100     FOBMAT(/1X,'**SOLVING LOAD FLOW EQUATIONS BY  NR**')

C       CALCULATING G AND H AT EACH BUS
        IF(LSET.EQ.1)GO TO 13
        DO 1 I=1,NBUS
        VMAG(I)=DBLE(VSPEC(I))
        VANG(I)=0.0D0
        P(I)=DBLE(PG(I))-DBLE(PD(I))
        Q(I)=DBLE(QG(I))-DBLE(QD(I))
1       CONTINUE
```

```
13      CONTINUE
        DO 7 I=1,NBUS
        CALP(I)=0.0D0
        CALQ(I)=0.0D0

        DO 4 J=1,NBUS
        CALP(I)=CALP(I)+VMAG(I)*VMAG(J)*YMAG(I,J)
        1*DCOS(YANG(I,J)-VANG(I)+VANG(J))
        CALQ(I)=CALQ(I)-VMAG(I)*VMAG(J)*YMAG(I,J)
        1*DSIN(+YANG(I,J)-VANG(I)+VANG(J))
4       CONTINUE

        G(I)=P(I)-CALP(I)
        H(I)=Q(I)-CALQ(I)


7       CONTINUE
C       CHECK FOR CONVERGENCE(GMAX &HMAX.LT.TOL)

        GMAX=0.0D0
        HMAX=0.0D0

        DO 5 I=1,NBUS
        IF(GMAX.LT.DABS(G(I))) GMAX=DABS(G(I))
        IF(HMAX.LT.DABS(H(I))) HMAX=DABS(H(I))
5       CONTINUE
        IF((GMAX.LE.TOL).AND.(HMAX.LE.TOL))GO TO 56

C**********GENERATING JACOBIAN AJ1
        DO 6 I=1,NBUS
        DO 8 J=1,NBUS
        IF(IABS (BTYPE(J)).EQ.3) GO TO 10.
        IF(J.EQ.I) GO TO 9
        AJ1(I,J)=-VMAG(I)*VMAG(J)*YMAG(I,J)
        1*DSIN(YANG(I,J)-VANG(I)+VANG(J))
        GO TO 8
9       AJ1(I,J)=0.0D0

        DO 12 K=1,NBUS
        IF (K.EQ.I) GO TO 12
        AJ1(I,J)=AJ1(I,J)+VMAG(I)*VMAG(K)*YMAG(I,K)
        1*DSIN(YANG(I,K)-VANG(I)+VANG(K))
12      CONTINUE

        GO TO 8
10      IF(J.EQ.I) GO TO 11
        AJ1(I,J)=0.0D0
        GO TO 8
11      AJ1(I,J)=-1.0D0

8       CONTINUE
6       CONTINUE

C**********GENERATING JACOBIAN AJ2
        DO 14 I=1,NBUS

        DO 17 J=1,NBUS
        IF(IABS (BTYPE(J)).GT.1) GO TO 18
        IF(J.EQ.I) GO TO 15
        AJ2(I,J)=VMAG(I)*YMAG(I,J)
        1*DCOS(YANG(I,J)-VANG(I)+VANG(J))
        GO TO 17

15      AJ2(I,J)=0.0D0
        DO 19 K=1,MBUS
        IF (K.NE.I) GO TO 16
        AJ2(I,J)=AJ2(I,J)+2*VMAG(I)*YMAG(I,I)
        1*DCOS(+YANG(I,I))
        GO TO 19
16      CONTINUE
        AJ2(I,J)=AJ2(I,J)+VMAG(K)*YEAG(I,K)
        1*DCOS(YANG(I,K)-VANG(I)+VANG(K))
19      CONTINUE
        GO TO 17

18      AJ2(I,J)=0.0D0

17      CONTINUE
14      CONTINUE
C**********GENERATING JACOBIAN AJ3
        DO 20 I=1,MBUS

        DO 24 J=1,MBUS
        IF(IABS (BTYPE(J)).EQ.3) GO TO 22
        IF(J.EQ.I) GO TO 23
        AJ3(I,J)=-VMAG(I)*VMAG(J)*YMAG(I,J)
        1*DCOS(YANG(I,J)-VANG(I)+VANG(J))

        GO TO 24
23      AJ3(I,J)=0.0D0

        DO 21 K=1,MBUS
        IF(K.EQ.J) GO TO 21
        AJ3(I,J)=AJ3(I,J)+VMAG(I)*VMAG(K)*YMAG(I,K)
        1*DCOS(YANG(I,K)-VANG(I)+VANG(K))
21      CONTINUE
        GO TO 24

22      AJ3(I,J)=0.0D0

24      CONTINUE
20      CONTINUE

C********** GENERATING JACOBIAN AJ4
        DO 29 I=1,MBUS

        DO 25 J=1,MBUS
        IF(IABS (BTYPE(J)).GT.1) GO TO 27
        IF(J.EQ.I) GO TO 28
        AJ4(I,J)=-VMAG(I)*YMAG(I,J)
        1*DSIN(YANG(I,J)-VANG(I)+VANG(J))
        GO TO 25

28      AJ4(I,J)=0.0D0

        DO 26 K=1,MBUS
        IF (K.NE.I) GO TO 30
        AJ4(I,J)=AJ4(I,J)-2*VMAG(I)*YMAG(I,I)
        1*DSIN(+YANG(I,I))
        GO TO 26
30      CONTINUE
```

```
        AJ4(I,J)=AJ4(I,J)-VMAG(K)*YMAG(I,K)
        1*DSIN(YANG(I,K)-VANG(I)+VANG(K))
26      CONTINUE
        GO TO 25
27      IF(J.EQ.I)GO TO 31
        AJ4(I,J)=0.0D0
        GC TO 25

31      AJ4(I,J)=-1.0D0
25       CONTINUE
29      CONTINUE

C       INVERTING JACOBIAN AJ4
        CALL MINV(AJ4,MBUS)


C***********GENERATING INVERSE SUBMATRIX BJ1,BJ2,BJ3,BJ4
C       BJ1=[AJ1-AJ2(AJ4**-1)AJ3]**-1

        CALL AMULT(XX,AJ2,AJ4,MBUS)
        CALL AMULT(YY,XX,AJ3,MBUS)
        DO 37 I=1,MBUS

        DC 39 J=1,MBUS
        BJ1(I,J) =AJ1(I,J)-YY(I,J)
39      CONTINUE
37      CONTINUE

C       INVERTING JACOBIAN BJ1
        CALL MINV(BJ1,MBUS)

C***********GENERATING JACOBIAN BJ2
C       BJ2=-BJ1*AJ2*(AJ4**-1)


        CALL AMULT(YY,BJ1,AJ2,MBUS)
        CALL AMULT (BJ2,YY,AJ4,MBUS)
        DO 44 I=1,MBUS
        DO 45 J=1,MBUS
        BJ2(I,J)=-1.0D0*(BJ2(I,J))
45      CONTINUE
44      CONTINUE

C***********GENERATING JACOBIAN BJ3
C       BJ3=-(AJ4**-1)*AJ3*BJ1
        CALL AMULT(YY,AJ3,BJ1,MBUS)
        CALL AMULT(BJ3,AJ4,YY,MBUS)

        DO 47 I=1,MBUS

        DO 48 J=1,MBUS
        BJ3(I,J)=-1.0D0*BJ3(I,J)

48      CONTINUE
47      CONTINUE

C***********GENERATING JACOBIAN BJ4
C       BJ4=(AJ4**-1)-(AJ4**-1)*AJ3*BJ2
        CALL AMULT(YY,AJ4,AJ3,MBUS)
```

```
          CALL AMULT(XX,YY,BJ2,NBUS)
          DO 51 I=1,NBUS
          DC 50 J=1,NBUS
          BJ4(I,J)=AJ4(I,J)-XX(I,J)
50        CONTINUE
51        CONTINUE

C         CALCULATING DELTA P,Q,VMAG,ANGLE
          DO 52 I=1,NBUS
          DP(I)=0.0D0
          DQ(I)=0.0D0
          DVMAG(I)=0.0D0
          DVANG(I)=0.0D0
          IF(IABS(BTYPE(I)).EQ.3)GO TO 54
          IF(IABS(BTYPE(I)).EQ.2)GO TO 55
          DC 53 J=1,NBUS
          DVMAG(I)=DVMAG(I)+BJ3(I,J)*G(J)+BJ4(I,J)*H(J)
53        DVANG(I)=DVANG(I)+BJ1(I,J)*G(J)+BJ2(I,J)*H(J)
          GO TO 52
54        DO 64 J=1,NBUS
          DP(I)=DP(I)+BJ1(I,J)*G(J)+BJ2(I,J)*H(J)

64        DQ(I)=DQ(I)+BJ3(I,J)*G(J)+BJ4(I,J)*H(J)
          GO TO 52
55        DO 65 J=1,NBUS
          DQ(I)=DQ(I)+BJ3(I,J)*G(J)+BJ4(I,J)*H(J)
65        DVANG(I)=DVANG(I)+BJ1(I,J)*G(J)+BJ2(I,J)*H(J)
52        CONTINUE


C         CALCULATING NEW P,Q,V,&ANGLE
          DO 62 I=1,NBUS
          P(I)=P(I)+DP(I)
          Q(I)=Q(I)+DQ(I)
          VMAG(I)=VMAG(I)+DVMAG(I)
          VANG(I)=VANG(I)+DVANG(I)
          IF(IABS(BTYPE(I)).NE.2)GO TO 62
          QMAX=QGMAX(I)-QD(I)
          QMIN=QGMIN(I)-QD(I)
          IF(Q(I).LT.DBLE(QMAX))GO TO 58
          Q(I)=DBLE(QMAX)
          BTYPE(I)=ISIGN(1,BTYPE(I))
          IBT(I)=2
          TYPE 3000,I
3000      FORMAT(1X,'BUS #',I3,2X,'EXCEEDS QGMAX,
          1BUS CHANGED TO TYPE 1')
          GO TO 62
58        IF(Q(I).GT.DBLE(QMIN)) GO TO 62
          Q(I)=DBLE(QMIN)
          BTYPE(I)=ISIGN(1,BTYPE(I))
          IBT(I)=2
          TYPE 3001,I
3001      FORMAT(1X,'BUS #',I3,2X,'EXCEEDS QGMIN,
          1BUS CHANGED TO TYPE 1')
62        CONTINUE
          LL=LL+1
          IF(LL.LT.LITER)GO TO 13

          TYPE 2300,LL
2300      FORMAT(1X,'****WILL NOT CONVERGE IN',I3,2X,' ITERATIONS')
          ABORT=1
```

```
        GO TO 60
C       PRINT OUT SOLUTIONS
56      TYPE 2400,LL
2400    FORMAT(1X,'****NR CONVERGENCE IN ',I3,' ITERATIONS')
        DO 70 I=1,MBUS
        PG(I)=P(I)+PD(I)
70      QG(I)=Q(I)+QD(I)
60      RETURN
        END


        SUBROUTINE GSLF
CSUBROUTINE TO CALCULATE LOAD FLOW EQUATIONS BY GAUSS-SEIDEL

        INTEGER MBUS,ABORT,BTYPE
        DIMENSION AI(15),BI(15,15)

        DOUBLE PRECISION AJ1,AJ2,AJ3,AJ4,BJ1,BJ2,BJ3,BJ4,
        1CALP,CALQ,DQ,DVANG,DVMAG,G,H,ITL,P,Q,VANG,VMAG,
        2XX,YANG,YMAG,YY,YYSHT,ZANG,ZMAG,YSER,
        3SMAG(15),SANG(15),AI,BI,VI,VII,VNEW,DELV,DVMAX,NANG
        COMMON/ /       AJ1(15,15),     AJ2(15,15),     AJ3(15,15),
        1AJ4(15,15),    ALPHA(15),      BDAT,           BETA(15),
        1BGSOL,         BJ1(15,15),     BJ2(15,15),     BJ3(15,15),
        1BJ4(15,15),    BTYPE(15),      BUSM,           CALP(15),
        1CALQ(15),      CB,CC,            CG,             CHB,
        1CHG,           CHL,            CL,             CHY,
        1CHZ,           DQ(15),         DVANG(15),      DVMAG(15),
        1EM(30),        ESP1,           ESP2,           G(15),
        1GAMMA(15),     GDAT,           H(15),          IC(15),
        1II,            ITER,           ITL(15),        KINV,
        1KSCL,          LAMBDA,         LDAT,           LITER,
        1LNGTH(30),     LNO,            LOSS,           LPSOL,
        1LSET,          MBUS,           N5,             NSWGB,
        1P(15),         PD(15),         PG(15),         PGMAX(15),
        1PGMIN(15),     Q(15),          QD(15),QDAT,        QG(15),
        1QGMAX(15),     QGMIN(15),      SB(30),         YYSHT(30,2),
        1TOL,TLDAT,     TPD,TBDA,      TPL,TGDAT,       VANG(15),
        1VMAG(15),      VSPEC(15),      XX(15,15),      YANG(15,15),
        1YMAG(15,15),   YSHT(30,2),     YY(15,15),      ZANG(15,15),
        1ZMAG(15,15),   ZSER(30,2),     YSER(30,2),     ABORT,IBT(15)

        TYPE·1000
1000    FORMAT(/1X,'**SOLVING LOAD FLOW EQUATIONS BY GS**')

C CALCULATING COMPLEX BUS POWER AND VARIABLES A & B
        DO 10 I=1,MBUS
        IF(LSET.EQ.1)GO TO 5
        VMAG(I)=DBLE(VSPEC(I))
        VANG(I)=0.0D0
5       IF(IABS(BTYPE(I)).NE.3)GO TO 12
        NSWGB=I
        GO TO 10
12      P(I)=DBLE(PG(I))-DBLE(PD(I))
        IF(IABS(BTYPE(I)).EQ.2)GO TO 25
        Q(I)=DBLE(QG(I))-DBLE(QD(I))
        SMAG(I)=DSQRT(P(I)*P(I)+Q(I)*Q(I))
        IF(P(I).EQ.(0.0D0))GO TO 14
        SANG(I)=DATAN2(Q(I),P(I))
        GO TO 22
14      IF(Q(I))16,18,20
16      SANG(I)=-1.570796326794897
        GO TO 22
18      SANG(I)=0.0D0
        GO TO 22
20      SANG(I)=+1.570796326794897
22      AI(I)=SMAG(I)/YMAG(I,I)
25      DO 10 J=1,MBUS
        BI(I,J)=YMAG(I,J)/YMAG(I,I)
10      CONTINUE

        L=0
28      DVMAX=0.0D0


C CALCULATING BUS VOLTAGES
        DO 30 I=1,MBUS
        IF(IABS(BTYPE(I)).EQ.3)GO TO 30
40      IF(IABS(BTYPE(I)).NE.2)GO TO 80


C TYPE 2 BUS VMAG IS RESET TO VSPEC, Q IS COMPUTED AND
C VARIABLES AI & AII ARE RECOMPUTED.
        Q(I)=0.0D0
        QMAX=QGMAX(I)-QD(I)
        QMIN=QGMIN(I)-QD(I)
        DO 60 J=1,MBUS
60      Q(I)=Q(I)-DBLE(VSPEC(I))*VMAG(J)*YMAG(I,J)*
        1DSIN(YANG(I,J)-VANG(I)+VANG(J))
        IF(Q(I).LE.DBLE(QMAX))GO TO 70
        Q(I)=DBLE(QMAX)
        BTYPE(I)=ISIGN(1,BTYPE(I))
        IBT(I)=2
        TYPE 3000,I
3000    FORMAT(1X,'BUS #',I3,2X,'EXCEEDS QGMAX,
        1BUS CHANGED TO TYPE 1')
        GO TO 72
70      IF(Q(I).GE.DBLE(QMIN))GO TO 72
        Q(I)=DBLE(QMIN)
        BTYPE(I)=ISIGN(1,BTYPE(I))
        IBT(I)=2
        TYPE 3001,I
3001    FORMAT(1X,'BUS #',I3,2X,'EXCEEDS QGMIN,
        1BUS CHANGED TO TYPE 1')
72      SMAG(I)=DSQRT(P(I)*P(I)+Q(I)*Q(I))
        IF(P(I).EQ.(0.0D0))GO TO 74
        SANG(I)=DATAN2(Q(I),P(I))
        GO TO 84
74      IF(Q(I))76,78,82
76      SANG(I)=-1.570796326794897
        GO TO 84
78      SANG(I)=0.0D0
        GO TO 84
82      SANG(I)=+1.570796326794897
84      AI(I)=SMAG(I)/YMAG(I,I)

C CALCULATING COMPLEX BUS VOLTAGES
80      VI=(AI(I)/VMAG(I))*DCOS(VANG(I)-YANG(I,I)-SANG(I))
        VII=(AI(I)/VMAG(I))*DSIN(VANG(I)-YANG(I,I)-SANG(I))
        DO 90 J=1,MBUS
```

```
      IF(J.EQ.I)GO TO 90
      VI=VI-BI(I,J)*VMAG(J)*DCOS(VANG(J)
      1+YANG(I,J)-YANG(I,I))
      VII=VII-BI(I,J)*VMAG(J)*DSIN(VANG(J)
      1+YANG(I,J)-YANG(I,I))
90    CONTINUE
      VNEW=DSQRT((VI*VI)+(VII*VII))
      IF(VNEW.EQ.0.0D0)GO TO 92
      IF(VI.EQ.(0.0D0))GO TO 86
      NANG=DATAN2(VII,VI)
      GO TO 95
86    IF(VII)88,92,94
88    NANG=-1.570796326794897
      GO TO 95
92    NANG=0.0D0
      GO TO 95
94    NANG=+1.570796326794897

C CALCULATING VOLTAGE MAGNITUDE DIFFERENCE BETWEEN ITERATIONS
C AND MULTIPLY BY ACCELERATION FACTOR OF 1.5
95    VNEW=VMAG(I)+1.5D0*(VNEW-VMAG(I))
      NANG=VANG(I)+1.5D0*(NANG-VANG(I))
      DELV=VNEW-VMAG(I)
      IF(DABS(DELV).GT.DVMAX)DVMAX=DABS(DELV)
      IF(IABS(BTYPE(I)).EQ.2)GO TO 96
      VMAG(I)=VNEW
96    VANG(I)=NANG
30    CONTINUE
      IF(DVMAX.LE.TOL)GO TO 100
      L=L+1
      IF(L.LE.LITER)GO TO 28
      TYPE 1010,L
1010  FORMAT(11X,'NO CONVERGENCE IN ',I3,2X,'ITERATIONS')
      ABORT=1
      RETURN

C COMPUTE SWING BUS POWER
100   I=NSWGB
      P(I)=0.0D0
      Q(I)=0.0D0
      DO 110 J=1,NBUS
      P(I)=P(I)+VMAG(I)*VMAG(J)*YMAG(I,J)*
      1DCOS(YANG(I,J)-VANG(I)+VANG(J))
      Q(I)=Q(I)-VMAG(I)*VMAG(J)*YMAG(I,J)*
      1DSIN(YANG(I,J)-VANG(I)+VANG(J))
110   CONTINUE
      TYPE 1020,L
1020  FORMAT(11X,'GSLF CONVERGENCE IN',I3,2X,'ITERATIONS')
      DO 120 I=1,NBUS
      PG(I)=P(I)+PD(I)
120   QG(I)=Q(I)+QD(I)
      RETURN
      END


      SUBROUTINE OPT
C SUBROUTINE TO CALCULATE ITL,IC, AND POWER FOR EACH GENERATOR
C CALCULATES OPTIMUM LAMBDA FOR SYSTEM

      REAL IC,LAMBDA
```

```
        INTEGER ABORT,BTYPE,MBUS
        DIMENSION PGSAV(15)

        DOUBLE PRECISION AJ1,AJ2,AJ3,AJ4,BJ1,BJ2,BJ3,BJ4,
        1CALP,CALQ,DQ,DVANG,DVMAG,G,H,ITL,P,Q,VANG,VMAG,
        2XX,YANG,YMAG,YY,YYSHT,ZANG,ZMAG,YSER,A,E
        COMMON/ /       AJ1(15,15),    AJ2(15,15),    AJ3(15,15),
        1AJ4(15,15),    ALPHA(15),     BDAT,          BETA(15),
        1BGSOL,         BJ1(15,15),    BJ2(15,15),    BJ3(15,15),
        1BJ4(15,15),    BTYPE(15),     BUSM,          CALP(15),
        1CALQ(15),      CB,CC,            CG,            CMB,
        1CMG,           CHL,           CL,            CMY,
        1CEZ,           DQ(15),        DVANG(15),     DVMAG(15),
        1EB(30),        ESP1,          ESP2,          G(15),
        1GAMMA(15),     GDAT,          H(15),         IC(15),
        1II,            ITER,          ITL(15),       KINV,
        1KSOL,          LAMBDA,        LDAT,          LITER,
        1LNGTH(30),     LNO,           LOSS,          LPSOL,
        1LSET,          MBUS,          N5,            NSWGB,
        1P(15),         PD(15),        PG(15),        PGMAX(15),
        1PGMIN(15),     Q(15),         QD(15),QDAT,       QG(15),
        1QGMAX(15),     QGMIN(15),     SB(30),        YYSHT(30,2),
        1TOL,TLDAT,     TPD,TBDAT,     TPL,TGDAT,     VANG(15),
        1VMAG(15),      VSPEC(15),     XX(15,15),     YANG(15,15),
        1YMAG(15,15),   YSHT(30,2),    YY(15,15),     ZANG(15,15),
        1ZMAG(15,15),   ZSER(30,2),    YSER(30,2),    ABORT,IBT(15)

        II=0
5       L=0
        TPD=0.0
        TPL=0.0
        TPGCC=0.0

C COMPUTE TOTAL POWER DEMAND & LOSS AND TOTAL POWER
C GENERATED AT CONSTANT COST
        DO 10 I=1,MBUS
        TPD=TPD+PD(I)
        TPL=TPL+P(I)
        IF(BTYPE(I).GE.0)GO TO 10
        IF((ALPHA(I).EQ.0.0).AND.(BETA(I).EQ.0.0).AND.
        1(GAMMA(I).EQ.0.0).EQ.0.0)TPGCC=TPGCC+PGMAX(I)
10      CONTINUE

C CHECK FOR NEGATIVE TOTAL POWER LOSS
        IF(TPL.GE.0.0)GO TO 20
        TYPE 1000
1000    FORMAT(11X,'SYSTEM POWER LOSS IS NEGATIVE.'/11X,' CHECK
        1TOTAL SYSTEM POWER GENERATION')
        RETURN

20      IF(LOSS.EQ.'Y')TPL=0.0
        IF(TPGCC.LT.(TPD+TPL))GO TO 30
        TYPE 1010
1010    FORMAT(11X,'MAX CONSTANT COST POWER NOT USED.'/11X,
        1'LOAD FLOW PROBLEM ONLY')
        RETURN

C DETERMINE INITIAL LAMBDA
30      IF(LSET.EQ.1)GO TO 40
        LAMBDA=5000.0
        DO 50 I=1,MBUS
```

```
      IF((BTYPE(I).GE.0).OR.(ALPHA(I).EQ.0))GO TO 50
      IF(ALPHA(I).LT.LAMBDA)LAMBDA=ALPHA(I)
      PGSAV(I)=PG(I)
50    CONTINUE
      DLAM=0.8*LAMBDA
      LSET=1
40    TYPE 1020
1020  FORMAT(5X,'***OPTIMIZING COST FUNCTIONS***')

C SKIP ITL CALCULATIONS IF LOSSLESS SYSTEM
      IF(LOSS.EQ.'Y')GO TO 60

C CALCULATING ITL'S
      DO 70 I=1,NBUS
      IF(BTYPE(I).GE.0)GO TO 70
      ITL(I)=0.0D0
      DO 80 J=1,NBUS
      IF(BTYPE(J).GE.0)GO TO 80
      A=((ZMAG(I,J)*DCOS(ZANG(I,J)))/(VMAG(I)*VMAG(J)))*
     1DCOS(VANG(I)-VANG(J))
      B=((ZMAG(I,J)*DCOS(ZANG(I,J)))/(VMAG(I)*VMAG(J)))*
     1DSIN(VANG(I)-VANG(J))
      ITL(I)=ITL(I)+(P(J)*A-Q(J)*B)
80    CONTINUE
      ITL(I)=ITL(I)+ITL(I)
70    CONTINUE

C OPTIMIZING LAMBDA
60    DPSAV=0.0
85    CONTINUE
      TPG=0.0
      DO 90 I=1,NBUS
      IF(BTYPE(I).GE.0)GO TO 90
      IF((ALPHA(I).EQ.0.0).AND.(BETA(I).EQ.0.0).AND.
     1(GAMMA(I).EQ.0.0))GO TO 100
      IF(LOSS.EQ.'N')GO TO 110
      IC(I)=LAMBDA
      GO TO 120
110   IC(I)=LAMBDA*(1.0-ITL(I))
      IF(IC(I).GE.ALPHA(I))GO TO 120
      PG(I)=0.0
      GO TO 89

C COMPUTE PG'S FROM COST FUNCTIONS
120   IF(GAMMA(I).NE.0.0)GO TO 130
      PG(I)=(IC(I)-ALPHA(I))/BETA(I)
      GO TO 89
130   C=BETA(I)*BETA(I)-4.0*(ALPHA(I)-IC(I))*GAMMA(I)
      IF(C.GE.0.0)GO TO 140
C     TYPE 1030,I
1030  FORMAT(1X,'COST CURVE COEFFICIENTS GIVE NEGATIVE
     1DETERMINANT,GENERATOR',I3,2X,'SET TO ZERO')
135   PG(I)=0.0
      GO TO 89
140   C=SQRT(C)
      PG(I)=(-BETA(I)+C)/(2.0*GAMMA(I))
      IF(PG(I).GE.PGMIN(I))GO TO 89
      PG(I)=(-BETA(I)-C)/(2.0*GAMMA(I))
      IF(PG(I).GE.PGMIN(I))GO TO 89
      PG(I)=PGMIN(I)
C     TYPE 1032,I
```

```
1032      FORMAT(1X,I3,1X,'PG NEG, GEN. SET TO 0.0')
          GO TO 89
100       PG(I)=PGMAX(I)
          IF((ALPHA(I).EQ.0.0).AND.(BETA(I).EQ.0.0).AND.
          1(GAMMA(I).EQ.0.0))GO TO 150
89        CONTINUE

C CHECK FOR PGMAX CONSTRAINTS
          IF(PG(I).LE.PGMAX(I))GO TO 105
          PG(I)=PGMAX(I)
C         TYPE 1036,I,PGMAX(I)
1036      FORMAT(1X,I3,'PG EXCEEDS PGMAX, GEN RESET TO',F12.7)
          GO TO 165
105       IF(PG(I).GE.PGMIN(I))GO TO 150
          PG(I)=PGMIN(I)
C         TYPE 1038,I,PGMIN(I)
1038      FORMAT(1X,I3,1X,'PG EXCEEDS PGMIN, GEN RESET TO',F12.5)
165       IF(GAMMA(I).NE.0.0)GO TO 160
          IC(I)=ALPHA(I)+BETA(I)*PG(I)
          GO TO 150
160       IC(I)=ALPHA(I)+BETA(I)*PG(I)+GAMMA(I)*PG(I)*PG(I)
150       TPG=TPG+PG(I)
90        CONTINUE

C CHECK POWER BALANCE LESS THAN ESP1
          DP=TPG-TPD-TPL
          IF(ABS(DP).LE.ESP1)GO TO 210
          L=L+1
          IF(L.LE.ITER)GO TO 170
          TYPE 1040,L
1040      FORMAT(11X,'POWER BALANCE EQUATIONS NOT SATISFIED IN'
          1,I3,2X,'ITERATIONS,'/11X,'CHECK COST CURVES AND
          2TOLERANCES')
          RETURN

C ADJUST LAMBDA IN COMPARISON TO PREVIOUS ITERATION
170       IF((DPSAV.GE.0.0).AND.(DP.GT.0.0))GO TO 180
          IF((DPSAV.LE.0.0).AND.(DP.LT.0.0))GO TO 190
          IF((DPSAV.GE.0.0).AND.(DP.LT.0.0))GO TO 200
          DLAM=0.5*DLAM
180       LAMBDA=LAMBDA-DLAM
          DPSAV=DP
          GO TO 85
200       DLAM=0.5*DLAM
190       LAMBDA=LAMBDA+DLAM
          DPSAV=DP
          GO TO 85
210       II=II+1
          TYPE 1050,L
1050      FORMAT(11X,'LAMBDA DETERMINED IN ',I3,2X,'ITERATIONS')

C CHECK PG(I) CONVERGENCE .LE. ESP2
          DO 220 I=1,NBUS
          IF(BTYPE(I).GE.0)GO TO 220
          IF(ABS(PG(I)-PGSAV(I)).GT.ESP2)GO TO 230
220       CONTINUE
          RETURN

230       DO 240 I=1,NBUS
          PGSAV(I)=PG(I)
          P(I)=DBLE(PG(I))-DBLE(PD(I))
          IF(IBT(I).NE.2)GO TO 240
          BTYPE(I)=ISIGN(2,BTYPE(I))
          VMAG(I)=1.0
          IBT(I)=0
240       CONTINUE
          IF(KSOL.EQ.1)GO TO 260
250       CALL NRLF
          GO TO 5
260       CALL GSLF
          GO TO 5
          END


         SUBROUTINE CTRI(A,B,C,D,N)
         DOUBLE PRECISION A(15,15),B(15,15),C(15,15),D(15,15),T(2),SUM(2)

C  TRIANGULARIZE A INTO B (LOWER) AND
C      B  UPPER TRIAG MATRIX
         DO 100 I=1,N
          C(I,1)=A(I,1)
100       D(I,1)=B(I,1)
         DO 105 J=2,N
         CALL CDIV(T(1),T(2),A(1,J),
          1B(1,J),A(1,1),B(1,1))
         C(1,J)=T(1)
105      D(1,J)=T(2)

         DO 275 IJ=2,N
         DO 250 J=IJ,N
         DO 250 I=IJ,N
         IF(I.LT.J) GO TO 205
         JM1=J-1
         SUM(2)=0.0D0
         SUM(1)=SUM(2)
         DO 200 K=1,JM1
         CALL CMUL(T(1),T(2),C(I,K),
        1 D(I,K),C(K,J),D(K,J))
         SUM(1)=SUM(1)+T(1)
200      SUM(2)=SUM(2)+T(2)
         C(I,J)=A(I,J)-SUM(1)
         D(I,J)=B(I,J)-SUM(2)
         GO TO 250
205      CONTINUE
         CALL CDIV(T(1),T(2),C(J,I),
        1 D(J,I),C(I,I),D(I,I))
         C(I,J)=T(1)
         D(I,J)=T(2)
250      CONTINUE
275      CONTINUE

         DO 280 I=1,N
         DO 280 J=1,N
         IF(I.GE.J) GO TO 280
         D(I,J)=0.0D0
          C(I,J)=D(I,J)
280      CONTINUE
         RETURN
         END
```

```
      SUBROUTINE CTRIV (A,B,C,D,N)
      DOUBLE PRECISION A(15,15),B(15,15),C(15,15),D(15,15),
     1SUM(2),T(2),AI

      I=N
10    CONTINUE
      K=I
20    CONTINUE
      SUM(2)=0.0D0
      SUM(1)=SUM(2)
      AI=SUM(1)
      IF(K-I) 25,22,25
22    AI=1.0D0
25    J=K+1
      IF(J.GT.N) GO TO 40

      DO 30 JJ=J,N
      CALL CMUL(T(1),T(2),A(JJ,K),
     1 B(JJ,K),C(I,JJ),D(I,JJ))
      SUM(1)=SUM(1)+T(1)
30    SUM(2)=SUM(2)+T(2)
40    T(1)=AI-SUM(1)
      T(2)=-SUM(2)
      CALL CDIV(SUM(1),SUM(2),T(1),T(2)
     1,A(K,K),B(K,K))
      C(K,I)=SUM(1)
      C(I,K)=C(K,I)
      D(K,I)=SUM(2)
      D(I,K)=D(K,I)

      K=K-1
      IF(K.EQ.0) GO TO 50
      GO TO 20
50    I=I-1
      IF(I.EQ.0) GO TO 60
      GO TO 10
60    CONTINUE

      RETURN

      END


       SUBROUTINE SOL

C SUBROUTINE TO CALCULATE AND PRINT BUS & GENERATOR SOLUTIONS
C      AND LINE POWER FLOW

       REAL LAMBDA,IC
       INTEGER BTYPE,BGSOL,EB,SB
       DOUBLE PRECISION AJ1,AJ2,AJ3,AJ4,BJ1,BJ2,BJ3,BJ4,
       1CALP,CALQ,DQ,DVANG,DVMAG,G,H,ITL,P,Q,VANG,VMAG,
       2XX,YANG,YMAG,YY,YYSHT,ZANG,ZMAG,YSER,
       3YA,YM,YSHTA,YSHTM,PLM,QLM


       COMMON/ /       AJ1(15,15),    AJ2(15,15),    AJ3(15,15),
       1AJ4(15,15),    ALPHA(15),     BDAT,          BETA(15),
             1BGSOL,          BJ1(15,15),     BJ2(15,15),   BJ3(15,15),
             1BJ4(15,15),     BTYPE(15),      BUSN,         CALP(15),
             1CALQ(15),       CB,CC,            CG,            CHB,
             1CHG,            CHL,            CL,           CMY,
             1CMZ,            DQ(15),         DVANG(15),    DVMAG(15),
             1EB(30),         ESP1,           ESP2,         G(15),
             1GAMMA(15),      GDAT,           H(15),        IC(15),
             1II,             ITER,           ITL(15),      KINV,
             1KSOL,           LAMBDA,         LDAT,         LITER,
             1LNGTH(30),      LNO,            LOSS,         LPSCL,
             1LSET,           MBUS,           N5,           NSWGB,
             1P(15),          PD(15),         PG(15),       PGMAX(15),
             1PGMIN(15),      Q(15),          QD(15),QDAT,      QG(15),
             1QGMAX(15),      QGMIN(15),      SB(30),       YYSET(30,2),
             1TOL,TLDAT,      TPD,TEDAT,      TPL,TGDAT,    VANG(15),
             1VMAG(15),       VSPEC(15),      XX(15,15),    YANG(15,15),
             1YMAG(15,15),    YSHT(30,2),     YY(15,15),    ZANG(15,15),
             1ZMAG(15,15),    ZSER(30,2),     YSER(30,2),    ABORT

             IF(BGSOL.EQ.0)GO TO 280
             IF(KINV.EQ.1)GO TO 265
             TPD=0.0
             TPL=0.0
             DO 275 I=1,MBUS
             TPD=TPD+PD(I)
             TPL=TPL+P(I)
      275    CONTINUE
             WRITE(N5,1157)TPL,TPD
      1157   FORMAT(/1X,'****SYSTEM BUS AND GENERATCR SOLUTIONS****'/
             11X,'SYSTEM LOSSES(PU)=',F8.4,',  SYSTEM LOAD(PU)=',F8.4,/
             21X,'BUS #',1X,'BTYPE',6X,'PG',6X,'QG',6X,'PD',6X,
             3'QD',5X,'VMAG',4X,'VANG')
             DO 277 I=1,MBUS
             PG(I)=P(I)+PD(I)
             QG(I)=Q(I)+QD(I)
      277    WRITE(N5,1158)I,BTYPE(I),PG(I),QG(I),PD(I),QD(I),
             1VMAG(I),VANG(I)
      1158   FORMAT(1X,I3,3X,I3,4X,6F8.4)
             BGSOL=0
             GO TO 280
      265    WRITE(N5,1155)II,ESP1,ESP2,LAMBDA,TPL,TPD
      1155   FORMAT(/1X,'****SYSTEM BUS AND GENERATOR SOLUTIONS****'/
             16X,'ECON. DISPATCH CONVERGENCE IN',I3,' ITERATIONS'/
             12X,'ESP1=',F8.4,', ESP2=',F8.4,',  LAMBDA=',F6.4,/1X,'SYSTEM
             3LOSSES(PU)=',F8.4,',  SYSTEM LOAD(PU)=',F8.4,/
             41X,'BUS#',1X,'BTYPE',4X,'IC',5X,'ITL',6X,'PG',6X,
             5'QG',6X,'PD',6X,'QD',5X,'VMAG',4X,'VANG')

             DO 270 I=1,MBUS
             PG(I)=P(I)+PD(I)
             QG(I)=Q(I)+QD(I)
      270    WRITE(N5,1160)I,BTYPE(I),IC(I),ITL(I),PG(I),QG(I),
             1PD(I),QD(I),VMAG(I),VANG(I)
      1160   FORMAT(1X,I3,1X,I3,F10.4,7F8.4)
             BGSOL=0
      280    CONTINUE
             IF(LPSOL.EQ.0)GO TO 300

      C CALCULATING LINE POWER FLOW
              WRITE(N5,1190)
```

```
1190     FORMAT(/1X,'****LINE POWER FLOW****'/1X,'LINE#',3X,'SB',4X,'EB',
         14X,'REAL POWER',4X,'REACTIVE POWER')
         DO 300 I=1,LNO
         L=SB(I)
         M=EB(I)
         YM=DSQRT((YSER(I,1)*YSER(I,1))+(YSER(I,2)*YSER(I,2)))
         YSHTM=DSQRT((YYSHT(I,1)*YYSHT(I,1))
         1+(YYSHT(I,2)*YYSHT(I,2)))
         IF(YSER(I,1).EQ.0.0D0)GO TO 302
         YA=DATAN2(YSER(I,2),YSER(I,1))
         GC TO 306
302      IF(YSER(I,2))303,304,305
303      YA=-1.570796326794897
         GO TO 306
304      YA=0.0D0
         GO TO 306
305      YA=1.570796326794897
306      IF(YYSHT(I,1).EQ.0.0D0)GO TO 308
         YSHTA=DATAN2(YYSHT(I,2),YYSHT(I,1))
         GO TO 314
308      IF(YYSHT(I,2))309,311,312
309      YSHTA=-1.570796326794897
         GO TO 314
311      YSHTA=0.0D0
         GO TO 314
312      YSHTA=1.570796326794897
314      CCNTINUE

         JJ=0
310      PLM=VMAG(L)*VMAG(L)*YM*DCOS(YA)
         1-VMAG(L)*VMAG(M)*YM*DCOS(VANG(L)-VANG(M)-YA)
         2+VMAG(L)*VMAG(L)*YSHTM*DCOS(YSHTA)

         QLM=-VMAG(L)*VMAG(L)*YM*DSIN(YA)
         1-VMAG(L)*VMAG(M)*YM*DSIN(VANG(L)-VANG(M)-YA)
         2-VMAG(L)*VMAG(L)*YSHTM*DSIN(YSHTA)
         WRITE(N5,1200)I,L,M,PLM,QLM
1200     FORMAT(1X,I3,4X,I3,3X,I3,F12.4,6X,F12.4)
         IF(JJ.EQ.1)GO TO 300
         JJ=1
         L=EB(I)
         M=SB(I)
         GO TO 310
300      CONTINUE
         LPSOL=0
         RETURN
         END


         SUBROUTINE AMULT(Z,X,Y,M)
         DOUBLE PRECISION  Z(15,15),X(15,15),Y(15,15)

         DO 1 I=1,M
         DO 1 J=1,M
         Z(I,J)=0.0D0
         DO 1 K=1,M
         Z(I,J)=Z(I,J)+X(I,K)*Y(K,J)
1        CONTINUE
         RETURN
         END
```

```
      SUBROUTINE CMUL(X,Y,A,B,C,D)
      DOUBLE PRECISION X,Y,A,B,C,D
      X=A*C-B*D
      Y=A*D+B*C
      RETURN
      END
      SUBROUTINE CDIV(X,Y,A,B,C,D)
      DOUBLE PRECISION X,Y,A,B,C,D
      X=C*C+D*D
      Y=(B*C-A*D)/X
      X=(A*C+B*D)/X
      RETURN
      END


      SUBROUTINE CINVT(A,B,C,D)
      A=C/(C*C+D*D)
      B=-D/(C*C+D*D)
      RETURN
      END


      FUNCTION CABS(A,B)
      DOUBLE PRECISION CABS,A,B,DSQRT
      CABS=DSQRT(A*A+B*B)
      RETURN
      END


      SUBROUTINE DPINVT(A,B,C,D)
      DOUBLE PRECISION A,B,C,D
      A=C/(C*C+D*D)
      B=-D/(C*C+D*D)
      RETURN
      END


      SUBROUTINE MINV (A,N)
C     THIS SUBPROGRAM IS FOR MATRIX INVERSION AND SIMUL
C     LINEAR EQUATIONS
C     A=THE GIVEN COEFFICIENT MATRIX A;A**-1 WILL BE
C     STORED IN THIS MATRIX AT RETURN TO THE MAIN PROGRAM
C
C     N=ORDER OF A;N.GE.1
      DIMENSION  IPVOT(15),INDEX(15,2)
      DOUBLE PRECISION A(15,15),B(15,15),PIVOT(15),T
      EQUIVALENCE (IROW,JROW),(ICOL,JCOL)
C
C     FOLLOWING 3 STATEMENTS FOR INITIALIZATION
      M=0
57    DET=1.0
      DO 17 J=1,N
17    IPVOT(J)=0
C
      DO 135 I=1,N
C
```

```
C       FOLLOWING 12 STATEMENTS FOR SEARCH FOR
C       PIVOT ELEMENT
C
        T=0.0D0
        DO 9 J=1,N
        IF(IPVOT(J).EQ.1) GO TO 9
13      DO 23 K=1,N
        IF(IPVOT(K)-1) 43,23,81
43      IF(DABS(T).GE.DABS(A(J,K)))  GO TO 23
83      IROW=J
        ICOL=K
        T=A(J,K)
23      CONTINUE
9       CONTINUE
C
        IPVOT(ICOL)=IPVOT(ICOL)+1
C
C       FOLLOWING 15 STATEMENTS TO PUT PIVOT
C       ELEMENT ON DIAGONAL
C
        IF(IROW.EQ.ICOL) GO TO 109
73      DET = -DET
        DO 12 L=1,N
        T=A(IROW,L)
        A(IROW,L) = A(ICOL,L)
12      A(ICOL,L) = T
        IF(M.LE.0) GO TO 109
33      DO 2 L=1,M
        T=B(IROW,L)
        B(IROW,L)=B(ICOL,L)
2       B(ICOL,L)=T
109     INDEX(I,1)=IROW
        INDEX(I,2)=ICOL
        PIVOT(I)=A(ICOL,ICOL)
        DET=DET*PIVOT(I)
C
C       FOLLOWING 6 STATEMENTS TO DIVIDE PIVOT ROW
C       BY PIVOT ELEMENT
C
        A(ICOL,ICOL)=1.0D0
        DO 205 L=1,N
205     A(ICOL,L)=A(ICOL,L)/PIVOT(I)
        IF (M.LE.0) GO TO 347
66      DO 52 L=1,M
52      B(ICOL,L)=B(ICOL,L)/PIVOT(I)
C
C       FOLLOWING 10 STATMENTS TO REDUCE NON-PIVOT ROWS
C
C
347     DO 135 LI=1,N
        IF(LI.EQ.ICOL)  GO TO 135
21      T=A(LI,ICOL)
        A(LI,ICOL)=0.0D0
        DO 89 L=1,N
89      A(LI,L)=A(LI,L)-A(ICCL,L)*T
        IF(M.LE.0) GO TO 135
18      DO 68 L=1,M
68      B(LI,L)=B(LI,L)-B(ICOL,L)*T
135     CONTINUE
C
C       FOLLOWING 11 STATEMENTS TO INTERCHANGE CONLUMS
```

```
C
222       DO 3 I=1,N
          L=N-I+1
          IF(INDEX(L,1).EQ.INDEX(L,2)) GO TO 3
19        JROW = INDEX(L,1)
          JCOL = INDEX(L,2)
          DO 549 K=1,N
          T = A(K,JROW)
          A(K,JROW)=A(K,JCOL)
          A(K,JCOL)=T
549       CONTINUE
3         CONTINUE
61        RETURN
          END
```

REFERENCES

Carpentier, J. 1962a. Contribution a l'Etude du Dispatching Economique, Bulletin de la Societe Francaise de Electricians, Series 8, Vol. 3 (August).

__________. 1962b. L'Optimisation de la Production a l'Electricite de Frane, Bulletin de la Societe Francoise de Electricians, Series 8, Vol. 3 (August).

Dommel, Hermann W., and William F. Tinney. 1968. Optimal Power Flow Solutions, IEEE Transactions on Power Apparatus and Systems, Vol. PAS-87, No. 10 (October).

Elgerd, Olle I. 1971. Electric Energy Systems Theory: An Introduction. New York, McGraw-Hill, Inc..

Friedlander, Gordon D. 1977. 20th Steam Power Station Cost Survey, Electrical World. New York: McGraw-Hill, Inc. (November).

George, E. E. 1943. Intrasystem Transmission Losses. A.I.E.E. Transactions, Vol. 62 (March):153-158.

Gross, George, and James F. Luini. 1971. Effecting Control of the Newton Load Flow, Proceedings of the 7th PICA (May).

Happ, H. H. 1977. Optimal Power Dispatch--A Comprehensive Survey, IEEE Transactions on Power Apparatus and Systems, Vol. PAS-26, No. 3 (May/June):841-853.

Happ, H. H., W. B. Ille, and R. H. Reisinger. 1962. Economic Systems Operation Considering Valve Throttling Losses, Part I--Method of Computing Valve-Loop Host Rates on Multi-Valve Turbines, AIEE Transactions, Vol. 81, Part III:609-615.

Kuhn, H. W., and A. W. Tucker. 1951. Non-Linear Programming, Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability, Berkeley & Los Angeles, University of California Press:481-492.

Martinez, Ralph. 1973. Users' Manual for Economic Dispatch System Program (EDSP), Computer Science Research Laboratory, CRSL Memo No. 245, University of Arizona (May).

Morrill, D. C., and J. A. Blake. 1955. A Computer Economic Scheduling and Control of Power Systems, A.I.E.E. Transactions, Vol. 74, Part III:1136-1141.

Ogbuobiri, E. C. 1970. Dynamic Storage and Retrieval in Sparsity Programming, IEEE Transactions on Power Apparatus and Systems, Vol. PAS-89:150-155.

The Oil and Gas Journal. May 1, 1978:32; June 12, 1978: 100-101.

Peschon, John, D. S. Piercy, W. F. Tinney, Odd J. Tveit, and Michael Cuenod. 1968. Optimum Control of Reactive Power Flow, IEEE Transactions on Power Apparatus and Systems, Vol. PAS-87 (January):40-48.

Peterson, Norris M., and W. Scott Meyer. 1971. Automatic Adjustment of Transformer and Phase-Shifter Taps in Newton Power Flow, IEEE Transactions on Power Apparatus and Systems, Vol. PAS-90, No. 1 (January/ February):103-108.

Ringlee, R. J., and D. D. Williams. 1962. Economic System Operation Considering Valve Throttling Losses, Part III--Distribution of System Loads by the Method of Dynamic Programming, AIEE Transactions, Vol. 81, Part III:615-622.

Sasson, Albert M., F. Abeytes, R. Cardenas, F. Gomes, and F. Vilora. 1971. A Comparison of Power Systems Static Optimization Techniques, Proceedings 7th PICA, Boston, Mass. (May):329-337.

Sasson, Albert M., and Hyde M. Merrill. 1974. Some Applications of Optimization Techniques to Power System Problems, Proceedings of IEEE, Vol. 62, No. 7 (July).

Skrotzki, Bernhardt G. A., and William A. Vopat. 1960. Power Station Engineering and Economy. New York, McGraw-Hill Book Company, Inc.

Stagg, Glenn W., and Ahmed H. El-Abiad. 1968. Computer Methods in Power System Analysis. New York, McGraw-Hill Book Company, Inc.

Steinberg, M. S., and T. H. Smith. 1943. Economic Loading of Power Plants and Electrical Systems. John Wiley & Sons, Inc.

Stott, Brian. 1974. Review of Load Flow Calculation Methods, Proceedings on IEEE, Vol. 62, No. 7 (July):916-929.

Subramanian, D. K., and H. N. Ramachandra Rao. 1969. Computation of Load Flows--A Comparison of Methods, Proceedings of 6th PICA, Denver, Colo. (May):584-591.

Tinney, W. F., and Clifford E. Hart. 1967. Power Flow Solutions by Newton's Methods, IEEE Transactions on Power Apparatus and Systems, Vol. PAS-86, No. 11 (November).

Tinney, W. F., and W. S. Meyer. 1973. Solution of Sparse Systems by Ordered Triangular Factorization. IEEE Transactions on Automatic Controls, Vol. AC-18, No. 4 (August).

Tinney, W. F., and Nobuo Sato. 1963. Techniques for Exploiting the Sparsity of the Network Admittance Matrix, IEEE Transactions on Power Apparatus and Systems, Vol. PAS-82 (December):944.

Tinney, W. F., and John W. Walker. 1967. Direct Solutions of Sparse Network Equations by Optimally Ordered Triangular Factorization, Proceedings of IEEE, Vol. 55, No. 11 (November):1801-1809.

Van Ness, James E. 1959. Iteration Methods for Digital Load Flow Studies, Transactions of AIEE, Part III (Power Apparatus and Systems), Vol. 75 (June): 583-588.

Ward, J. B., and H. W. Hale. 1956. Digital Computer Solutions of Power Flow Problems, Transactions of AIEE, Vol. 75, Part II:398-404.