8523090

Enouen, Paul William

A COMBINED TRANSMISSION-DISTRIBUTION LOAD FLOW MODEL EMPLOYING SYSTEM REDUCTION AND VOLTAGE VARIABLE LOAD REPRESENTATION

The University of Oklahoma PH.D. 1985

University
Microfilms
International 300 N. Zeeb Road, Ann Arbor, MI 48106

PLEASE NOTE:

In all cases this material has been filmed in the best possible way from the available copy. Problems encountered with this document have been identified here with a check mark __√__.

1. Glossy photographs or pages _____
2. Colored illustrations, paper or print ______
3. Photographs with dark background _____
4. Illustrations are poor copy ______
5. Pages with black marks, not original copy ______
6. Print shows through as there is text on both sides of page ______
7. Indistinct, broken or small print on several pages ______
8. Print exceeds margin requirements ______
9. Tightly bound copy with print lost in spine ______
10. Computer printout pages with indistinct print __√__
11. Page(s) __________ lacking when material received, and not available from school or author.
12. Page(s) __________ seem to be missing in numbering only as text follows.
13. Two pages numbered ______. Text follows.
14. Curling and wrinkled pages ______
15. Dissertation contains pages with print at a slant, filmed as received ________
16. Other ____

THE UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

A COMBINED TRANSMISSION--DISTRIBUTION LOAD FLOW MODEL EMPLOYING SYSTEM REDUCTION AND VOLTAGE VARIABLE LOAD REPRESENTATION

A DISSERTATION

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

degree of

DOCTOR OF PHILOSOPHY

BY

PAUL WILLIAM ENOUEN

Norman, Oklahoma

1985

A COMBINED TRANSMISSION--DISTRIBUTION LOAD FLOW MODEL
EMPLOYING SYSTEM REDUCTION AND VOLTAGE
VARIABLE LOAD REPRESENTATION

APPROVED BY

DISSERTATION COMMITTEE

ACKNOWLEDGMENTS

The author wishes to recognize the support of Dr. John E. Fagan in the problem solving stage as well as in the preparation of this document. Dr. Fagan also served as committee chairman, an effort which is deeply appreciated.

Thanks are also due to Dr. M. E. Council, Dr. S. C. Lee, Dr. J. C. Thompson, and Dr. A. A. Aly for service on the graduate committee and for the encouragement and assistance rendered by each.

Dr. Emmanuel Annan offered friendship and support in the work, both of which deserve fond gratitude.

The consulting firm Joyce, Gray and Associates assisted by supplying system data used in the test of the model.

Pat Ybarra provided excellent typing service in the initial preparation of this volume, as did Janet Maginnis in the final stages.

Last but not least the author wishes to thank his wife, Jerrie, without whose encouragement, assistance and forebearance this work would never have been completed.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

A COMBINED TRANSMISSION--DISTRIBUTION LOAD FLOW MODEL
EMPLOYING SYSTEM REDUCTION AND VOLTAGE
VARIABLE LOAD REPRESENTATION

CHAPTER I

INTRODUCTION

In the few decades since its introduction, the digital computer has found widespread application within the electric power industry. One of the more fruitful areas for its utilization has been in the load flow calculation.

A successful load flow calculation provides a complete description of the state of the real and reactive powers in the system, under steady state condition, and with specified loads. This information is essential in evaluating the adequacy of a present or planned system. The effects of contingencies may be examined by altering the system data to reflect the abnormal configuration, before running the calculation.

Prior to the advent of large digital computers, power system engineers used AC calculating boards (network analyzers) to solve the load flow problem (1). This device used variable resistances, inductances and capacitances interconnected to form a miniature replica of the system. Network equivalents consisted of the pi equivalent of each transmission line, generator units which provided independent adjustment of voltage magnitude and phase angle, units to represent loads, transformer equivalent circuits and other device equivalents. The power supply for various boards was 60 to 10,000 Hz, most being designed for 440 or 480 Hz. Elaborate metering methods provided for measuring current, voltage, and real and reactive power at each unit. Setting up the connections, making adjustments and reading the data were tedious and time consuming. In addition, the accuracy of the results was limited by the precision of the settings and of the metering equipment. In 1960 some 50 AC calculating boards were in constant daily use in North America. The task has now been completely taken over by digital computers.

The Load Flow Problem

Mathematically, the load flow calculation is nothing more than a problem in circuit analysis. The difficulty arises from the fact that the number of nodes and lines may be in the thousands and the observed state variables cause the solution technique to be non-linear. Considerable insight may be gained however, by examining a small system, as in Figure 1.

This system has one generator connected at bus (node) 0, and loads connected at buses 1,2, and 3. The buses are interconnected by lines represented as impedances with subscripts to indicate endpoints. The magnitude of the voltage at the

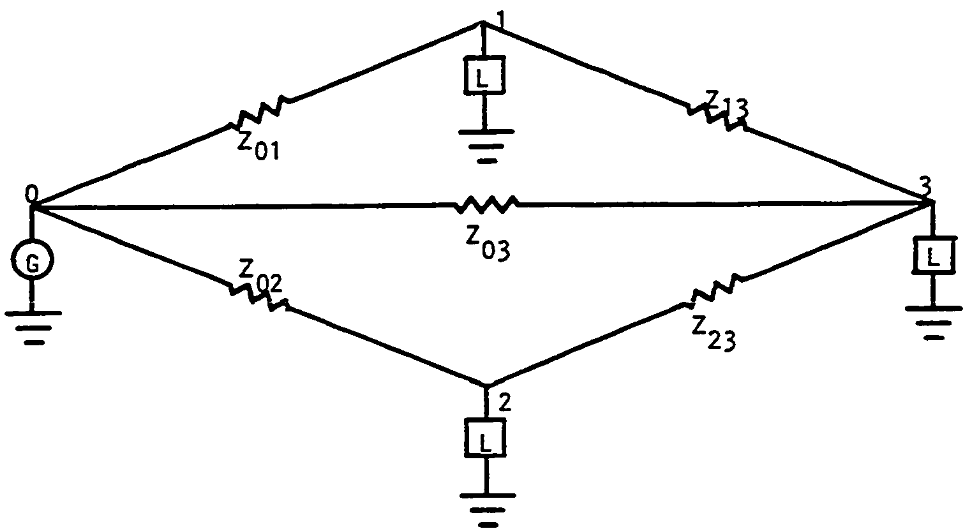


Figure 1. Small Example System.

generator bus is specified. The angle of the generator bus voltage is also specified, usually at zero degrees, and this is used as the reference angle for all other voltages and currents in the system. Some description of the loads is given; this will be covered in more detail later in this chapter and in Chapter III. The real and reactive power inputs from the generator are not given. The generator must satisfy the needs of the loads, and also supply any power lost in the system itself. Since these losses cannot be calculated until the final solution is obtained the actual generator input is unknown until that time. A generator bus with specified voltage magnitude and angle and unspecified real and reactive power input is called a "swing" bus, or occasionally a "slack" bus. If the voltage magnitudes and angles at each of the remaining buses can be found the calculation of all the currents and powers becomes trivial, and the problem is solved. The seed of the problem then is to find those voltage magnitudes and angles.

Early approaches to the digital computer solution of the load flow problem used the loop frame of reference in admittance form (2). The loop admittance matrix was obtained by a matrix inversion, a procedure which is both time consuming and costly. The specification of the network loops involved tedious data preparation, and the results, when obtained, were difficult to interpret. In addition, if a network was altered in any way, the tedious matrix inversion had to be repeated before another case could be run. For these reasons the method did not enjoy widespread use.

Later techniques used the bus frame of reference in the admittance form to describe the network. This method gained wide popularity because of the simplicity of data preparation and the ease with which the bus admittance matrix could be formed and modified for network changes in subsequent cases. To illustrate how readily a problem is set up in the bus frame of reference the equations for the example system in Figure 1 will now be written. In this effort the admittances of the connecting lines will be used instead of the impedances, and the loads will be assumed to be of the constant admittance type. Applying Kirchhoff's current law at each of the load buses gives:

$$(V_1 - V_o)Y_{10} + (V_1 - V_3)Y_{13} + V_1Y_{L1} = 0$$

$$(V_2 - V_o)Y_{20} + (V_2 - V_3)Y_{23} + V_2Y_{L2} = 0$$

$$(V_3 - V_o)Y_{30} + (V_3 - V_1)Y_{31} + (V_3 - V_2)Y_{32} + V_3Y_{L3} = 0$$

Expanding and collecting terms yields:

$$\begin{array}{lllll} (Y_{10} + Y_{13} + Y_{L1})V_1 & & -Y_{13}V_3 & = & Y_{10}V_o \\ & (Y_{20} + Y_{23} + Y_{L2})V_2 & -Y_{23}V_3 & = & Y_{20}V_o \\ -Y_{31}V_1 & -V_{32}V_2 & (Y_{30} + Y_{31} + Y_{32} + Y_{L3})V_3 & = & Y_{30}V_o \end{array}$$

In matrix form this is written

$$[I] = [Y][V]$$

where $[I]$ is the vector $\begin{bmatrix} Y_{10}V_o \\ Y_{20}V_o \\ Y_{30}V_o \end{bmatrix}$ the elements of which are the currents into the nodes and are constants; V is the vector $\begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix}$ the elements of which are the unknown voltages, and $[Y]$ is the matrix

$$\begin{bmatrix} (Y_{10} + Y_{13} + Y_{L1}) & 0 & -Y_{13} \\ 0 & (Y_{20} + Y_{23} + Y_{L2}) & -Y_{23} \\ -Y_{31} & -Y_{32} & (Y_{30} + Y_{31} + Y_{32} + Y_{L3}) \end{bmatrix}$$

all of the components of the matrices Y, V and I are complex quantities. The simplicity is obvious. The diagonal terms for a particular bus are just the sums of all the admittances connected to the bus, including the load admittance. The off-diagonal terms are the negative of the individual admittances in the lines leading from the bus. If line charging current is to be considered appropriate admittances are added to the diagonal terms for each node. This matrix is defined as the Bus Admittance Matrix (symbol Y_{bus}) (1). It can be constructed without going through the tedium of actually writing the equations, and is very easily done by the computer. If the network is altered, say by changing the wire size in the line from bus i to bus j, only four elements in Y_{bus} are affected. They are the elements Y_{ij} and Y_{ji}, which are the negative of the admittance of the line itself, and the elements Y_{ii} and Y_{jj}, the diagonal elements which represent buses i and j. Thus changes can be handled easily. The simplicity of the operations just described has led to the almost universal adoption of the bus frame of reference and the admittance form in load flow calculations.

Formulating the problem is one step. Solving the resulting set of simultaneous equations is quite another. Solution techniques will be discussed in the next chapter.

Historically, load flow calculations on transmission and distribution systems have been done separately. In fact the term "load flow" and the formulation described above normally apply only to the transmission systems. A similar calculation (i.e., solve for the node voltages) for a distribution system is called a "voltage profile". One of the reasons for the difference in approach is that the two types of systems are basically different. The transmission system contains many sources (generators) and sinks (loads) and is a mesh. Figure 2 is a one line diagram of a typical transmission system; the IEEE 39 bus test system. The numbered small circles represent generation stations. The heavier lines, also numbered, represent the buses, and are the nodes of the network. For simplicity all transformers have been omitted. Note that there are eight closed loops in the network, making it a mesh, and that there are seven more lines than nodes. The distribution system contains, only one source, the substation. Figure 3 is a one line diagram of a typical distribution system; again, all transformers have been omitted. The nodes are numbered and represent points where individual loads are connected. The heavy line at the left is the substation bus, and it could be any of the buses 1 thru 29 on figure 2. Note that there are no closed loops, making it a radial system, and that if the substation bus is not counted as a node, there are exactly as many nodes as lines.

Another significant difference is the manner in which loads are handled. Modern transmission load flow programs treat loads as some form of constant power or constant KVA, resulting in non-linear equations (see Chapter II). Distribution voltage profile programs on the other hand consider loads to be constant impedance, and linear solution techniques may be used.

The Contribution

The contribution to knowledge which will result from the efforts described herein will be embodied in the development of a load flow model that treats the transmission and distribution systems together. This will allow the ready calculation of, for example, the effect upon the transmission

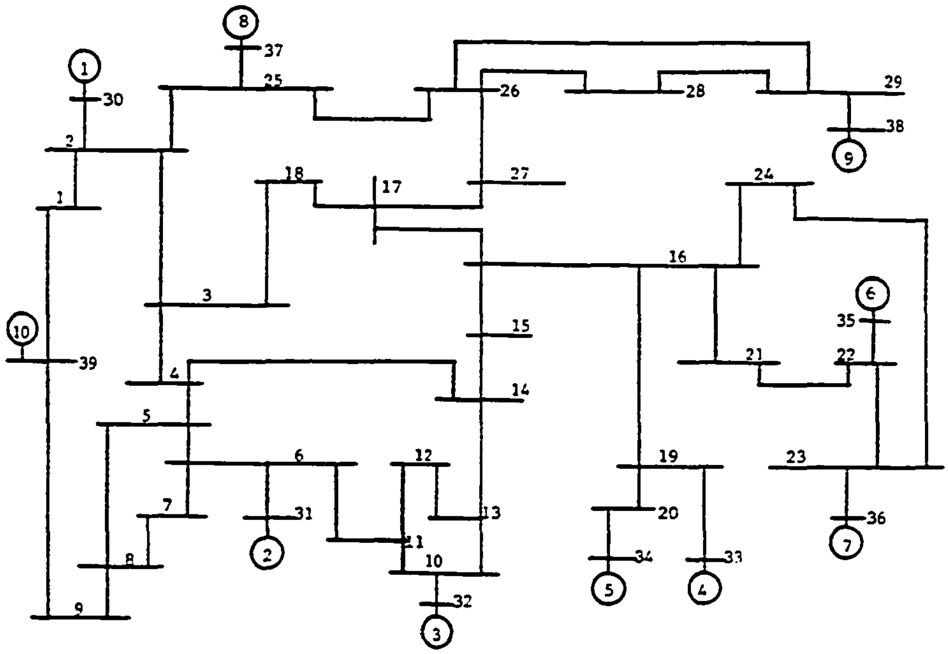


Figure 2. Typical Transmission System.

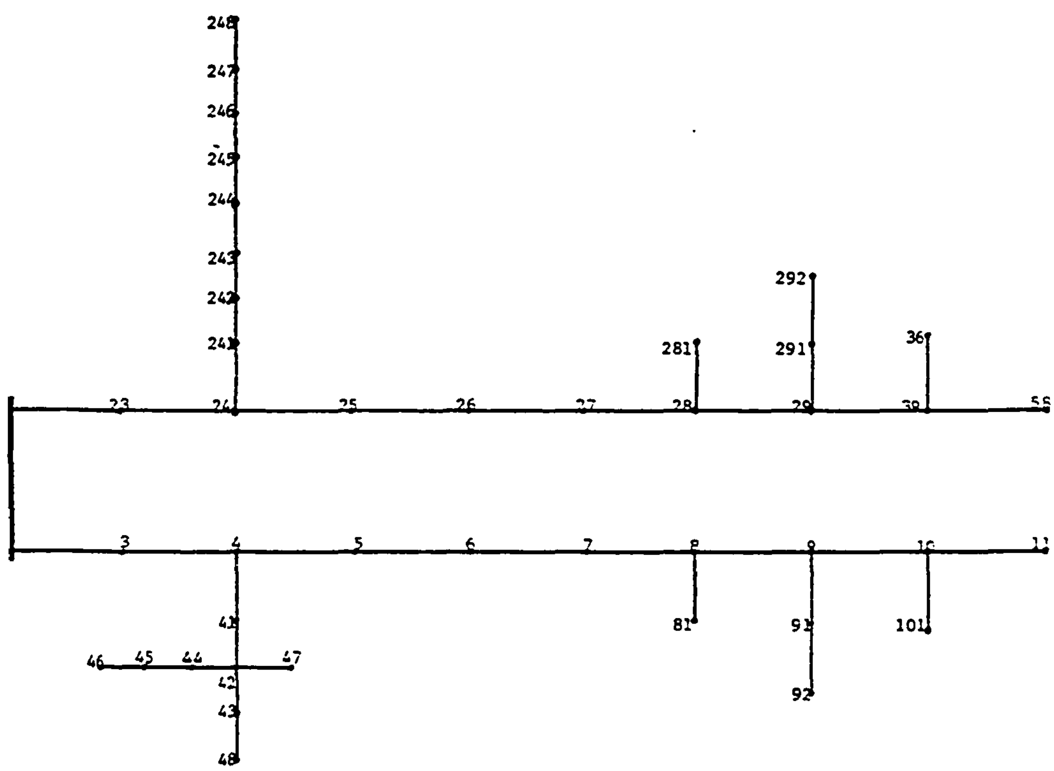


Figure 3. Typical Distribution System.

system of a large load added somewhere on the distribution system, and at the same time provide a voltage profile for the distribution system. Under the present procedure, the plan of attack would be first to add the new load to the already accumulated load for the substation (bus) selected, and run a load flow on the transmission system to predict the voltage at that substation; then use the predicted substation voltage in a voltage profile calculation for the distribution system, with the new load in place. In order to obtain a complete picture of the revised system, the engineer must go through five steps: 1) add the load at the substation, 2) Run the load flow, 3) Adjust the substation voltage for the voltage profile, 4) Connect the load to the distribution system, 5) Run the voltage profile. The new program will permit the reduction of the five steps to two: 1) Connect the load to the distribution system, 2) Run the program. The development of the model will require the completion of two subtasks:

1. The objective of the model is to calculate the impact of a distribution load upon the transmission system. If one were to combine the two systems in their entirety, the number of nodes would be very large, and the problem size would render it unmanageable. However, any effect of the distribution system upon the transmission system would be most pronounced at the bus that represents the distribution substation, and next at those buses adjacent to the substation.

A very good picture may be obtained by examining only that part of the transmission system which contains those buses of interest. The first sub-task then is to select and implement a reduction process to limit the size of the problem, yet preserve the effects of the circuit elements in the reduced portion. System reduction will be addressed more thoroughly in Chapter III.

2. Presently load flow calculations and voltage profile calculations use different load representations, neither of which is consistent with reality. The second sub-task is to determine a load representation which approaches the behavior of actual loads and lends itself to implementation in a load flow model. Load representation is covered in more detail in Chapter III.

In summary, the goal is to develop a load flow model to handle the transmission and distribution systems simultaneously, using a network reduction algorithm to keep the problem at a manageable size, and including a more realistic representation of the system loads.

Chapter Outline

Chapter II will examine the load flow problem and go into detail on many different techniques which have been used in its solution. Desirable and undesirable features of each will be cited.

Chapter III discusses the techniques and selects one to to be used in this research. Various approaches to system reduction and load representation are also examined.

Chapter IV presents the contribution of this thesis by describing the model that has been developed to solve the combined load flow problem; explaining the reduction technique that has been selected and the model segment that implements it; and then discusses an improved load representation, including how it fits into the model.

Chapter V describes the data input requirements of the new model, and explains how the different segments of the model tie together.

Chapter VI presents test results for two different sample problems.

Chapter VII contains conclusions and recommendations.

CHAPTER II

BACKGROUND

The network equations for a small power system were derived in Chapter I. The intent was to show how easily that could be done using the bus frame of reference in admittance form. The resulting equations cannot be used directly to solve the load flow problem, but they serve as the framework around which the load flow equations are built. Stott in (14) presents an analytical formulation of the load flow problem.

Problem Statement

The task in general is to find the voltage magnitude and angle at every node, given the real and reactive power requirements at the nodes. We are not explicitly interested in the currents. If I_i is the complex current into node i and V_i the complex voltage at the same node, then $V_i^* I_i = P_i - jQ_i$, where P and Q are real and reactive powers, respectively, j is the imaginary operator ($\sqrt{-1}$) and * represents complex conjugation. One line of the matrix equation $[I] = [Y][V]$ can be written

$$I_i = \sum_{k=1}^{n} Y_{ik} V_k .$$

Then pre-multiplying by V_i^* gives

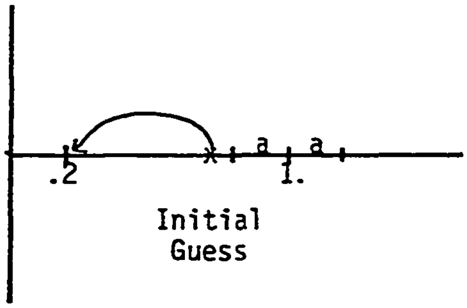
$$V_i^* I_i = P_i - jQ_i = V_i^* \sum_{k=1}^{n} Y_{ik} V_k$$

The equation is now written in terms of the quantities given and the quantities desired. P and Q are given and Y is constant. We must find the V's that satisfy the equation. Since two of the unknowns are always multiplied together ($V_i^* V_k$) the problem is non-linear and numerical methods must be used to find the solution. The technique selected may use a rearranged version of the equation or make some simplifying assumptions but the correct solution must satisfy this equation.

A numerical method begins by selecting initial values for all of the unknown quantities. These values are plugged into the equation to see if it is satisfied. If not, corrections to the values are made and it is tried again. This is repeated until the needs are met to within certain prespecified tolerances. If the initial values picked turn out to be very close to the solution values then the process should converge quickly to the proper solution. However, if the initial values are not close enough, or if the solution technique selected is weak, the process may not converge to a proper solution even though one may certainly exist. Since the problem is non-linear it is possible that more than one mathematically correct solution exists. It is not likely that more than one solution would be satisfactory from a practical standpoint. Figure 4 tries graphically to show convergence to an infeasible solution. The process may also fail to converge by continually oscillating about the true solution, or by diverging, as shown in Figure 5. In a load flow problem it is customary to set the initial values at 1.0 per unit magnitude (the nominal voltage) and 0.0 degrees.

As one examines the load flow problem, two features stand out. One is the shear size of the problem. Literally thousands of nodes and lines may be involved. Since we are dealing with complex quantities, for n nodes, the size of the complex matrix would be $(2n)^2$. The memory required to store this vast amount of data, and the computational burden presented are certainly limiting factors.

The second prominent feature is that, using the popular bus frame of reference in admittance form, most of the elements of the matrix are zero. The matrix is very sparse. Recalling the way the matrix was constructed in Chapter I, if a node has four line sections connected (a realistic average for a transmission system), it will contribute nine elements to the matrix; a diagonal element, four off-diagonal elements in the row, and four off-diagonal elements in the column. There would be about 9n non-zero complex numbers in the matrix for a system of n nodes. If the system is radial, like a distribution system, then the matrix is even more sparse. In order to be strictly radial, each node can have, on the average, just two lines connected; one in and one out. Thus a node contributes only five complex elements



Correct solution is 1.

Infeasible solution is .2

A guess of 1±a would have caused convergence to 1.

Figure 4. Convergence to Infeasible Solution.

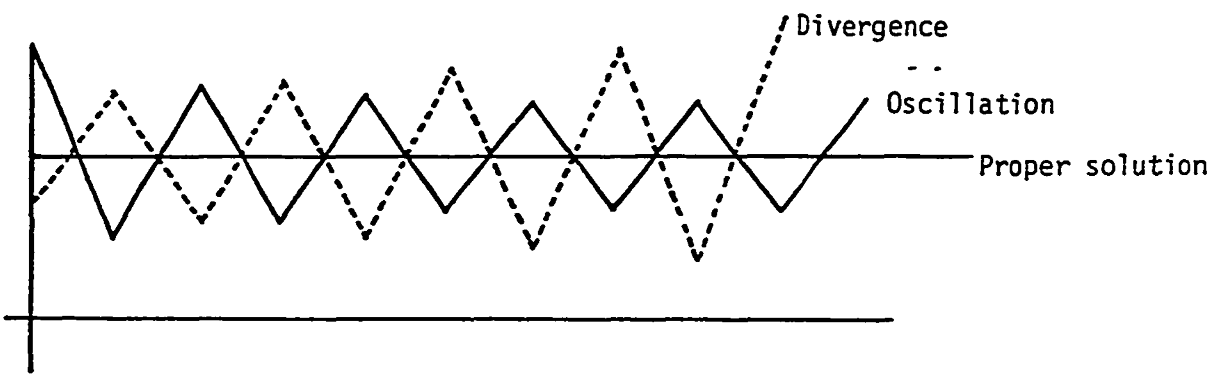


Figure 5. Failure to Converge.

to the matrix in a radial system. The objective of this thesis is to develop a load flow model that combines the transmission and distribution systems. The foregoing suggests that the matrix in such a problem would be more sparse than usual, which will have ramifications later on.

Solution Techniques

In the 25 or so years since the first application of the digital computer to the load flow problem, literally hundreds of papers have appeared (14) discussing the subject. From these, several ideas have gained wide acceptance in the community. Four of the basic techniques will be reviewed here. Several refinements will be looked at next, followed by a brief discussion of non-linear programming. A technique will be selected for use within this work.

In the consideration of a solution process, three items become important: Computation time, storage space and liklihood of convergence. The discussion to follow will include all of these.

Relaxation

The "Relaxation" technique solves for current differences ΔI at each bus by computing the current required at the bus by the load, and subtracting the calculated current into the bus; i.e., $\Delta I_i = \frac{P_i - jQ_i}{V_i^*} - \sum_{k=1}^{n} V_k Y_{ik}$. Then, at the bus with the largest ΔI, it adjusts the voltage E_i to eliminate ΔI; $\Delta E_i = - \frac{\Delta I_i}{Y_{ii}}$. It uses the new E_i and starts over again. When the ΔE_i become less than a specified figure convergence is assumed. The computation required at each iteration is simple and straight forward, though some time is lost searching for the largest ΔI. Convergence for large problems may require many iterations, increasing computer time accordingly. Only the non-zero terms of the Y matrix need be stored so that space requirements are near minimum. Convergence is governed by the Y matrix, and it will be discussed in detail in the second following section.

Gauss

This technique is similar to the previous one; it calculates directly the voltage at each bus using the power required at the bus, the voltage at connected busses and the admittance of attached lines. At each bus i

$$E_i = \frac{1}{Y_{ii}} \quad \frac{P_i - jQ_i}{E_i^*} - \sum_{\substack{k=1 \\ k \neq i}}^{n} E_k Y_{ik}$$

At the completion of each iteration, all the voltages are changed to the new values, and it starts again. When the voltage changes all become less than a specified value, convergence is assumed. There is no search here for the largest ΔI, so the computation time per iteration is less than in the relaxation methods. Slightly more storage is required to save the new voltages until it's time to change them all. Convergence characteristics are also similar to the relaxation method, and criteria will be discussed in the next section.

Gauss-Seidel

This is an improvement on the previous method. It uses the same equations, but here, when a voltage is calculated it is immediately inserted in place of the old voltage and used in all subsequent calculations. This removes the need to store both voltage vectors. In addition, since the latest data are used at each calculation step, convergence is reached in fewer iterations than with the previous techniques. The Gauss-Seidel method has enjoyed great popularity in the industry, but as the size of the problems continues to grow it is losing some of its appeal. For large systems the number of iterations required for convergence is on the order of n, the number of busses, and the total iterative computing time varies approximately with n^2 (14). This method, as well as the last two, is structurally based on the Y matrix, and it is the character of that matrix which determines convergence. Matrix theory shows that convergence is realized if the largest eigenvalue-modulus of the iteration matrix is less than unity (14). A more useful though over-stringent condition is that Y should possess strict diagonal dominance. Conditions on a power system can reduce the diagonal dominance and prevent convergence. These conditions include junctions of high and low impedances, and capacitors. The problem sometimes exists with transmission systems, which have several elements per row and column in the Y matrix. It would even more likely be encountered in a radial system, with an average of two off-diagonal elements per row or column.

Newton-Raphson

The Newton-Raphson method is supplanting Gauss-Seidel in the load flow picture. The techniqueuses first partial derivatives to calculate changes in voltage needed to correct errors in power at each bus. The real and reactive powers are treated separately, and the voltage is broken into either real and imaginary parts, or magnitude and angle, depending on the formulation to be used. This will be described in detail in a later part of this chapter. This treatment produces an array of partial derivatives (Jacobian) of dimension 2n. If all of these elements needed to be stored the space required would be prohibitive, but again, only the non-zero elements need be saved. The structure of each portion of the Jacobian is identical to the Y matrix; it is equally sparse. Total storage is greater since the Y matrix must be saved as well as the Jacobian, and the Jacobian contains approximately four times as many elements. Time per iteration is much greater for this method, due to the need to calculate a new Jacobian at each iteration, but the convergence is so quick that it usually beats the other techniques. Depending on the desired accuracy, Newton-Raphson usually converges in 2-5 iterations regardless of the size of the system. Therefore computation time varies with n rather than n^2, and it becomes more attractive for large systems. Convergence criteria are much less stringent with this method, with the most critical factor appearing to be the closeness of the initial values. From an analytical viewpoint there would seem to be no reason why this procedure would not perform just as well on a radial system.

Fast De-Coupled

This method (16) takes advantage of some of the characteristics of a transmission system and greatly simplifies the Newton-Raphson approach. Normally the relationship is very weak between the real power and the voltage magnitude, and also between the reactive power and the voltage angles. Here the relation is eliminated and the problem is treated as two separate blocks; real power vs. angle, and reactive power vs. magnitude. The blocks are iterated in turn. Storage is claimed to be 40% less than that needed for Newton-Raphson, and time for each iteration is also less. On the systems tested the method converged dependably, but took more iteration than expected by Newton-Raphson. For application to the present problem, the intital premise may present difficulty. The weak relations mentioned are partly due to the normally high X:R ratios of transmission lines; so high that in many analyses the resistance is ignored completely. In distribution lines the X:R ratio is much lower; in many cases less than unity. The effect would be to slow the convergence considerably, or even prevent it.

Second Order Techniques

Several methods have been developed (17), (18), (19) which use the second partial derivatives in the solution process. This is the equivalent of using the first three terms in the Taylor Expansion for the system, rather than the first two as in Newton-Raphson. Claims and counter-claims in the papers and discussions thereof serve to confuse the issue. Performances are compared to the Fast De-Coupled technique and to Newton-Raphson, with the second-order methods prevailing. They are said to converge more quickly and use only slightly greater storage; and to be more effective with ill-conditioned systems. Time per iteration is longer since the second order terms must be considered. The total solution time is probably about even when compared to Newton-Raphson. It is certainly not clear at this point that the second order techniques are universally superior. The Newton-Raphson approach is not yet in danger of eclipse.

Nonlinear Programming

Nonlinear programming techniques have been successfully applied to power system problems; specifically, in the minimum loss and economic dispatch areas. Sasson (20) used the Fletcher-Powell method to solve the load flow problem and investigated Fiacco-McCormick, Lootsma, and Zangwill for minimum loss and economic dispatch questions. He found that the Fletcher-Powell method was successful in some load flow problems in which the Gauss-Seidel method failed to converge. The solution time was comparable. Yu (21) used a Generalized Reduced Gradient technique to address a new formulation of the minimum loss load flow problem. He found it to be slower than the Gauss-Seidel approach.

CHAPTER III

THE MODEL

The load flow problem has been with us for as long as power systems have existed. It was not until the development of the digital computer that it actually became possible to solve the problem to a reasonable degree of accuracy for systems of any significant size. The application of the computer to the problem proceeded through several evolutionary steps until it finally settled into the bus frame of reference in admittance form. This formulation of the problem is so simple and so easily programmed into and handled by the computer that, in retrospect, it is difficult to see why any other formulation was even considered.

At the same time, computers themselves were advancing rapidly; growing in capacity and speed, making it ever more practical to treat larger and larger systems. Throughout this time however, the load flow problem considered only transmission systems. Distribution system problems were solved separately, and differently. The intent of this research is to develop a model which will handle the two types of systems at the same time and in the same way. The combining of the systems will cause changes in the structure of the problem which may significantly effect the behavior of the solution algorithm. The developmant of the model will begin by selecting from those techniques described in the previous chapter the method which seems to offer the greatest liklihood of success in the combined problem.

Selection of Technique

The objective of this thesis is to demonstrate that the transmission and distribution systems can reasonably be combined in a single load flow solution. With that end in mind, the most important of the three criteria cited at the start of this chapter is: probability of convergence. Though important, computation time, and to a greater extent, storage requirements can be effected by programming. With skillful programming there is no overwhelming advantage in these categories for any of the methods reviewed. It is not intended herein to expend great effort toward minimising either; but only to show that a combined solution can be attained. In cases where the probabilities of convergence are about the same, computation time and storage requirements may be used as tie-breakers.

In the first three techniques reviewed; Relaxation, Gauss, and Gauss-Seidel, the probability of convergence depends to a great extent on the degree of diagonal dominance in the Y matrix. The combined system to be treated will predominantly be radial. Consequently the diagonal dominance will be weakened, and the liklihood of convergence considerably reduced. It would not be practical to select one of these techniques.

The Newton-Raphson method has been found to converge for problems in the transmission system in which the previous three methods fail. Additionally, the probability of success is not effected by the character of the Y matrix. The structure of the combined problem doesn't present any difficulties. This method is not ruled out.

The Fast De-Doupled Load Flow has enjoyed considerable success with transmission system load flow problems. The combined system will bring with it different types of line admittances which substantially alter the justification for the de-coupling. The much lower X:R ratios in the distribution lines reduce the liklihood of convergence (22) and it seems that a de-coupled technique would not be a wise choice.

The second order methods do not suffer in convergence, in that they appear to be reliable. Nothing in the analysis suggests that the reliability would be lessened in a mostly radial system. As for computation time and storage, they appear to be on about the same level as Newton-Raphson. The formulation of the problem would be more difficult because of the need to calculate the second derivatives.

The non-linear programming approach cannot be ruled out on the basis of convergence. The techniques tested showed themselves to be reliable. Their weakest point is computational time. Yu (21) states that the technique he used was slower than Gauss-Seidel. In discussion of (20) Dy Liacco says, "As an end in itself, a load flow program using non-linear programming cannot compete with Newton's method, in our opinion. We do not think it can even compare closely."

The Newton-Raphson technique is selected. The convergence characteristics are more promising than with Relaxation, Gauss, Gauss-Seidel, or Fast De-Coupled. The problem formulation is simpler than with a second-order method. It is faster than non-linear programming. The selection is fortuitous for yet another reason. It is intended that a system reduction technique will be incorporated, as described later in the chapter. The most promising approach was designed for use with the Newton-Raphson load flow. The Newton-Raphson method will now be described.

Newton-Raphson

The Newton-Raphson approach was adapted to the load flow problem by Tinney and Hart (3). It starts with the equations:

$$I_i = \sum_{j=1}^{n} Y_{ij} V_j .$$

Pre-multiplying by V_i^* changes the equations to constant power form:

$$V_i^* I_i = P_i - jQ_i = V_i^* \sum_{j=1}^{n} Y_{ij} V_j \qquad (1)$$

Newtons method involves the repeated direct solution of a system of linear equations derived from equation (1). By Taylor's theorem, a function of x may be expanded about a point x_o as follows

$$f(x) = f(x_o) + \frac{df(x)}{dx}(x - x_o) + \frac{d^2f(x)}{dx^2}\frac{(x-x_o)^2}{2!} + \ldots$$

or

$$\Delta f = \frac{df}{dx}\Delta x + \frac{d^2f}{dx^2}\frac{(\Delta x)^2}{2!} + \ldots$$

If Δx is small, the terms including $(\Delta x)^2$ and higher powers may be ignored, leaving $\Delta f = \frac{df}{dx}\Delta x$. When the theorem is applied to a system of n simultaneous equations, and only the first order terms are considered, the result is:

$$[\Delta f] = [J][\Delta x]$$

where $[\Delta f]$ is the vector $[\Delta f_1, \Delta f_2, \ldots \Delta f_n]^T$,

$[\Delta x]$ is the vector $[\Delta x_1, \Delta x_2, \ldots \Delta x_n]^T$,

and $[J]$ is the Jacobian for the function f_i

$$[J] = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & & \frac{\partial f_2}{\partial x_n} \\ \vdots & & & \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & & \frac{\partial f_n}{\partial x_n} \end{bmatrix}$$

with each of these derivatives evaluated at the point x_o.

The Jacobian matrix of equation (1) gives the linearized relationship between small changes in voltage angle ($\Delta\delta_k$) and normalized magnitude ($\Delta E_k/E_k$), and small changes in power, (ΔP_k and ΔQ_k).

The linearized equations can be written in general:

$$\begin{bmatrix} \Delta P_k \\ \\ \Delta Q_k \end{bmatrix} = \begin{bmatrix} H_{ki} & N_{ki} \\ \\ J_{ki} & L_{ki} \end{bmatrix} \begin{bmatrix} \Delta\delta_k \\ \\ \dfrac{\Delta E_k}{E_k} \end{bmatrix} \begin{matrix} k = 1,n \\ i = 1,n \end{matrix} \qquad (2)$$

where

$$H_{ki} = \frac{\partial P_k}{\partial \delta_i}; \quad N_{ki} = \frac{\partial P_k E_i}{\partial E_i}; \quad J_{ki} = \frac{\partial Q_k}{\partial \delta_i}; \quad L_{ki} = \frac{\partial Q_k E_i}{\partial E_i}$$

The partial derivatives above are real functions of the admittance matrix and the node voltages.

The solution proceeds as follows:

(1) Select arbitrary values for each of the node voltages.

(2) Solve equations (1) for the resulting P and Q at each node.

(3) Since each node has a scheduled P and Q, the differences ΔP and ΔQ can be found by subtracting the resulting P and Q from the scheduled P and Q at each node.

(4) Compare ΔP and ΔQ with a desired maximum error. If any ΔP or ΔQ exceeds the maximum, proceed. If

all ΔP and ΔQ are less than the maximum, the voltage angles and magnitudes are considered solved.

(5) Calculate the elements of the Jacobian using the latest voltages.

(6) Solve equation (2) for $\Delta\delta$ and $\Delta E/E$ at each node, using the calculated ΔP and ΔQ.

(7) Adjust the voltage angles δ_i by $\Delta\delta_i$ and the magnitudes E_i by ΔE_i and return to step 2.

The convergence criteria for the Newton-Raphson method are less stringent than those of the Gauss-Seidel method. The initial guess (x_o) must be sufficiently near the final result (x) that the approximation made earlier, i.e., ignoring the terms in Δx of power 2 and greater, is reasonably valid. The convergence of the Newton-Raphson method will be examined in greater detail in Chapter III.

Radial System

In a radial distribution system the problem is much more easily solved. The loads are represented as constant impedances, which means that the equations are linear. The only source is a substation, which is usually handled as a constant voltage alone, or as a constant voltage behind a small impedance to account for voltage drop in the transmission system. A simple technique is to start at the far end and accumulate load and line impedances by series and parallel combinations working back toward the source. Then using the total impedance, find the current leaving the source. Use this current to calculate the voltage drop in the first line section, and thus the voltage at the next node. Use the node voltage to calculate the current to any load attached, and to the next line segments. This procedure continues to the far end of the feeder, at which time all voltages will be known. In the voltage drop calculation an approximation often used is:

$$V_d = IR\cos\theta + IX\sin\theta$$

where

I = Line current (magnitude)

R = Line resistance

X = Line reactance

θ = Angle between I and source voltage

The approximation simplifies the arithmetic, and the error introduced is not large enough to be significant in a normal distribution system.

Another technique starts with the assumption of 1.0 per unit voltage at all nodes. The KW and KVAR loads are then accumulated starting at the far end, working back toward the substation, and including line losses. Once at the substation the current can be calculated, and then the voltage drop in the first line section. As in the last method, these calculations are continued to the end of the feeder. The voltages thus calculated will be in error however, as they were found using load current based on 1.0 per unit volts at each node, a condition which no longer exists. Thus the process must be iterated using the most recently calculated voltages until the differences between iterations are less than some maximum.

It is seen that there are significant differences between the solution methods used for mesh systems and radial systems, and no effort has been noted to date to combine the two load flow problems into one. If it is desired to investigate the consequences of a particular distribution load allocation on the transmission system it is necessary first to analyze the distribution system, note the effects at the substations, and then run the transmission load flow using the noted conditions. If the cases to be examined are numerous, this procedure can quickly become cumbersome.

System Reduction

The shear size of the problem was quickly recognized as a severe limiting factor in the application of the digital computer to the load flow calculation. This has led to many and diverse efforts to circumvent the difficulty through some sort of problem modification. An interesting approach called "Diakoptics" was pioneered by Kron (4). It involved "tearing" the network into two or more parts, solving the smaller parts, and recombining the solved parts into a whole. The result is a solution of the entire system. Since in this present thesis what is sought is a solution of a small part of the system, the diakoptics approach is not the answer. What is needed is a way to permanently eliminate from all further consideration those parts of the system which are not of immediate interest, while preserving the effects of the eliminated part upon the retained part. Four methods of accomplishing this will be examined briefly here.

Star-Mesh Transformation

The simplest approach to a reduction of this type is to use star-mesh transformation, and series and parallel combinations to eliminate unwanted nodes. Unfortunately, this method cannot be applied to any source nodes, or to any nodes with non-linear loads attached, which severely limits its usefulness.

Classical Reduction

"Classical" reduction (5) proceeds from Kirchoff's current law in matrix form: I = YE. Allowing the subscript 1 to denote the subvector of voltages and currents to be reduced and the subscript 2 to denote those to be retained, the equation can be re-written

$$\begin{bmatrix} I_1 \\ I_2 \end{bmatrix} = \begin{bmatrix} Y_{11} & Y_{12} \\ Y_{21} & Y_{22} \end{bmatrix} \begin{bmatrix} E_1 \\ E_2 \end{bmatrix}$$

or in expanded form

$$I_1 = Y_{11}E_1 + Y_{12}E_2$$

$$I_2 = Y_{21}E_1 + Y_{22}E_2$$

Solving the first for E_1, substituting in the second and rearranging gives

$$[Y_{22} - Y_{21}Y_{11}^{-1}Y_{12}]E_2 = I_2 - Y_{21}Y_{11}^{-1}I_1$$

Define: $I_{2eQ} = I_2 - Y_{21}Y_{11}^{-1}I_1$ and

$$Y_{22eQ} = Y_{22} - Y_{21}Y_{11}^{-1}Y_{12}$$

Then

$$I_{2eQ} = Y_{22eQ}E_2$$

To be useful requires that Y_{11}, Y_{12}, Y_{21} and I_1 be known and constant. Since I_1 is the vector of current injections into the reduced portion it is unrealistic to assume it to be constant, and the technique loses some of its appeal.

REI

The REI net, of the radial (R) type, equivalent (E) for a node and independent (I) of the rest of the network preserves the identity of eliminated generators as controlled voltage sources (5,6). The generators in the reduced part are replaced by an equivalent generator. The complex power injected by the equivalent generator is the sum of all the original complex generator powers:

$$S_e = \sum S_{gi}$$

The remaining load buses are reduced using classical reduction techniques. The REI reduction overcomes some of the deficiencies in classical reduction, particularly those associated with generators in the reduced part.

Linearized Reduction

A technique called "Linearized Reduction" (5) operates on the Jacobian matrix of the portion of the system to be reduced. A Jacobian correction matrix is developed, the elements of which are then added to the appropriate elements of the Jacobian for the retained part. The same correction matrix is also applied to the matrix of powers which flow from the reduced part to the retained part. These corrections work to preserve the effects upon the retained part of conditions within the reduced part. The technique has been applied to investigate the effects of contingencies within the transmission system. A thorough discussion of this method, including the derivation of the equations will be provided in Chapter III.

Load Representation

Neither the constant KVA representation in the transmission load flow nor the constant impedance representation in the distribution voltage profile can accurately depict the behavior of all loads. Figure 6 shows the voltage-current characteristics of constant power and constant impedance loads. A constant current load is also shown for comparison. Incandescent lighting and electric heating loads are almost purely resistive in nature and therefore a constant impedance represents them very well. In that case:

$$I = \frac{V}{R} \quad \text{and} \quad P = VI = \frac{V^2}{R}.$$

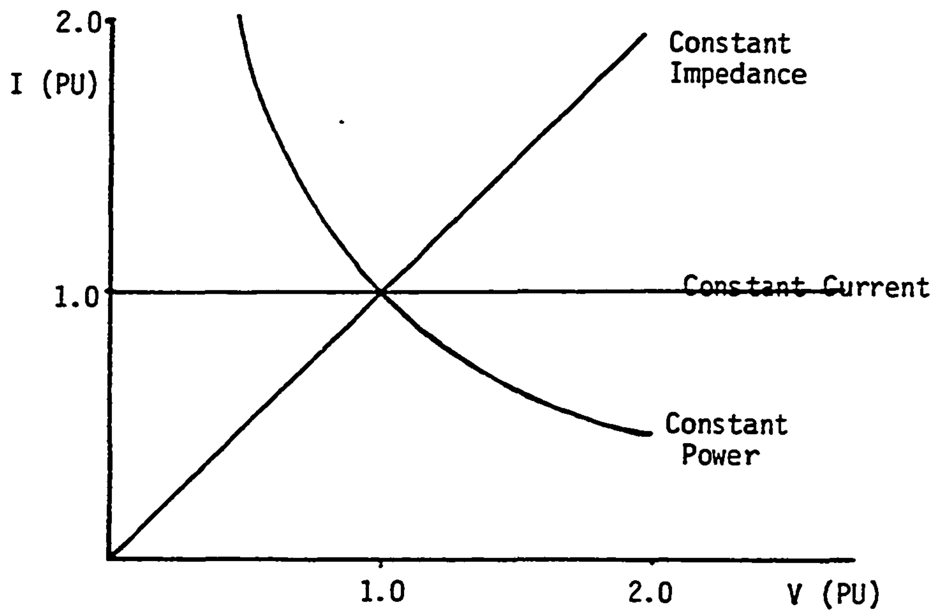


Figure 6. Load Representations

If a voltage-variable load representation is to be used to model incandescent lighting or electric heating, the voltage-squared term will predominate in the real power portion. also since the load is resistive, there should be no reactive power part.

An induction motor behaves much like a constant real power load in the voltage range of interest, i. e. 1.0 $\pm$ .05 P. U. That is: $P = VI = C$; and $I = \frac{C}{V}$. The reactive power may vary considerably with changes in voltage however. A well designed motor will operate at its best power factor at rated voltage and rated power. This suggests that as the voltage moves away from nominal in either direction, the reactive power will increase. There must be a term in the reactive power expression where the voltage has an exponent greater than unity.

It is apparent even from this brief discussion that a voltage-variable load must have at least two components; one for real power and one for reactive power.

Realistic loads as seen by a substation, or at a node in a distribution system would not simply be one or the other of these types. They would instead be combinations of these and other, more complicated loads, so that each of the components would be expected to be more involved than just a single term with an integer exponent. Such a combination of loads might be handled as a function of voltage by:

$P = P_o + P_1V + P_2V^2$ and

$Q = Q_o + Q_1V + Q_2V^2$ where

$P_o + P_1 + P_2 = 1$

$Q_o + Q_1 + Q_2 = 1$ and

P_o, P_1 and P_2, and Q_o, Q_1 and Q_2 are the portions of constant power, constant current and constant impedance real and reactive power respectively. This is called the "Quadratic Form" of load representation (7).

The "Single Exponential" form (7,8) is written:

$$P = P_o\left|\frac{E}{E_o}\right|^{k_P}$$

$$Q = Q_o \left| \frac{E}{E_o} \right|^{k_Q}$$

where K_P and K_Q are exponents which are varied to suit the load. This form is capable of providing representations equivalent to those of the quadratic form in the vicinity of the normal operating voltage.

The discovery of the proper values to assign to the p's and q's in the quadratic form, and to the k's of the exponential form, is no trivial task. Load characteristic information is quite diverse, and it is usually emphasized that specific determination requires specific investigation, possibly actual measurement.

This measurement is precisely what has been done in (9), not just for selected items of equipment, but for entire sections of a distribution system. The frequency and the voltage were varied; the real and reactive powers were measured, and the results provided mathematical formulae for the real and reactive powers as functions of frequency and voltage differences from nominal; i.e., Δf and ΔV. These findings will be examined more closely in the next chapter.

CHAPTER IV

MODEL IMPLEMENTATION

The total task can be divided into three fairly distinct steps. They are: 1) Construct a conventional load flow model; 2) Implement the system reduction algorithm; and 3) Include the new load representation. In this chapter each of these three steps will be treated in turn. The three computer programs which comprise the new model are included as appendices.

Load Flow

The decision was made to construct a basic model from scratch rather than try to adapt one that already existed. The reason was that the later steps, particularly the inclusion of the new load representation, would involve modification of the inner workings of the model. This would require an intimate familiarity with the structure and flow of the model; an intimacy which would best be gained through the actual construction itself. Another point to consider was the scope of the task; that is, only to prove that the concept works, not to develop a production grade model. Refinements could be added later as a separate project. The model would operate as follows:

1) The input, the output and all calculations would be handled in per unit quantities. The model itself would make no conversions. 2) Since finding the bus voltages amounts to solving the problem, these voltages are all that will be solved for explicitly. Line currents and power flows would not be calculated, nor would flags be included to indicate high or low bus voltages or overloaded lines.

The next decision to be made involved the selection of the technique to be used in solving the loadflow problem. The Newton-Raphson approach was chosen because of the overall characteristics displayed, as mentioned in Chapter II. Also, as will be seen in the next section, the system reduction technique operates on the Jacobian matrix, an entity which does not exist in the other methods.

In their landmark paper (3) Tinney and Hart described a load flow program using the Newton-Raphson method which was made practical by including sparse matrix techniques and optimally ordered Gaussian elimination. Their primary obstacle was the fact that the computer at their disposal had only 32 K of core memory available. The IBM 370/158 being used in the present effort provides, in the largest job class, 640 K of core storage. Therefore it seemed likely that a problem of a size large enough to verify the concept could be treated with straight forward techniques. It was decided then that the program would be simply written, and then tested on the IEEE 118 bus test system. If the simple program could not handle that system, then more sophisticated programming techniques would be employed. If the simple program succeeded with 118 bus test system, then the sophisticated techniques would be added to the list of possible refinements mentioned earlier.

The 118 bus test system was developed by IEEE to provide a common basis for the evaluation of load flow models. The configuration is such that it presents severe convergence problems, especially with reactive power. If a particular load flow model converges for the 118 bus test system, then it is likely to converge for any real system, and it is seen as a fair test of the concept under development here.

The Newton Raphson method is well covered in the literature. Carnahan (11) describes the procedure generally, while the other references cited (2,3,10) look only at the load flow application. The problem can be formulated in either rectangular or polar coordinates. Stagg (2) takes the former course, and Tinney and Hart (3) and Van Ness (10) the latter. The polar option was selected for use here because it treats voltage controlled buses in a simple manner, which will be described later.

The Newton-Raphson method was briefly discussed in Chapter II, and it was shown that the linearized equations can be written in general:

$$\begin{bmatrix} \Delta P_k \\ \\ \Delta Q_k \end{bmatrix} = \begin{bmatrix} H_{ki} & N_{ki} \\ \\ J_{ki} & L_{ki} \end{bmatrix} \begin{bmatrix} \Delta \delta_k \\ \\ \dfrac{\Delta |E_k|}{|E_k|} \end{bmatrix} \quad \begin{matrix} k = 1,N \\ i = 1,N \end{matrix} \tag{3}$$

where

$$H_{ki} = \frac{\partial P_k}{\partial \delta_i}\,; \quad N_{ki} = \frac{\partial P_k |E_i|}{\partial |E_i|}; \quad J_{ki} = \frac{\partial Q_k}{\partial \delta_i}; \quad L_{ki} = \frac{\partial Q_k |E_i|}{\partial |E_i|}$$

Derivations of the equations for the partial derivatives above were given by Van Ness (10), and the equations were then used by Tinney and Hart (3). They are repeated here.

For the off-diagonal terms:

$$H_{km} = L_{km} = a_m f_k - b_m e_k, \qquad m \neq k$$

$$N_{km} = -J_{km} = a_m e_k + b_m f_k, \qquad m \neq k$$

where $e_k + jf_k = E_k \angle\delta$, the voltage at bus k,

$G_{km} + jB_{km} = Y_{km} \angle\theta$, the admittance connecting busses k and m,

and $a_m + jb_m = (e_m + jf_m)(G_{km} + jB_{km})$, the current at bus m contributed by bus k.

The rectangular forms are used here to simplify the expressions. Even so, the polar option is still being used, since the partial derivatives are taken with respect to voltage angle and magnitude.

For the diagonal terms:

$$H_{kk} = -Q_k - B_{kk}|E_k|^2$$

$$L_{kk} = Q_k - B_{kk}|E_k|^2$$

$$N_{kk} = P_k + G_{kk}|E_k|^2$$

$$J_{kk} = P_k - G_{kk}|E_k|^2$$

where P_k is the calculated net real power at bus k, and Q_k is the calculated net reactive power at bus k.

Three types of buses are considered, swing bus, load bus and voltage controlled bus. The swing bus is described in chapter I as a generator bus at which the voltage magnitude and angle are both specified, and the real and reactive powers are not specified. Only one bus is so designated. ΔP_k and ΔQ_k are the differences, or mismatches between the specified real and reactive powers and the calculated real and reactive powers respectively, at bus k. Since at a swing bus neither P nor Q is specified, the quantities ΔP and ΔQ are meaningless, and the swing bus contributes no equations to the linearized system. Also, since both $|E|$ and δ are fixed, partial derivatives with respect to these quantities will not exist, and the swing bus will contribute no terms to the other equations in the linearized system.

At a load bus the real and reactive power are both specified, but the voltage angle and magnitude are not. At these buses ΔP and ΔQ are both meaningful, and hence a load bus contributes two equations to the linearized system. And since the voltage angle and magnitude are both permitted to change, a load bus contributes two terms to each equation in the system, H and N terms to each ΔP equation, and J and

L terms to each ΔQ equation.

At a voltage controlled bus the voltage magnitude is specified, as is the real power. The voltage angle is permitted to change, and the reactive power is not specified. Thus only the ΔP is meaningful, providing one equation. The changing voltage angle contributes one term to each equation, H terms to ΔP equations and J terms to ΔQ equations. The behavior of a voltage controlled bus presupposes the existence of a reactive power source or sink at the bus to accomodate the reactive power calculated to be there. This source or sink must have limits, and these limits are provided to the model as upper and lower bounds of reactive power capability. Once the problem has converged, the reactive power calculated for each voltage controlled bus is compared to the limits. If a limit is exceeded, the reactive power required at the bus is set to the value of that limit, the voltage is set free to vary, and the problem is restarted. The bus has become a load bus and is treated as such for the remainder of the problem. This changes the structure of the Jacobian by adding a ΔQ equation, and also by adding another term to each existing equation.

In order to calculate the elements of the Jacobian a voltage magnitude and angle must be available for each bus. To start the program, each of these items which has not already been specified is provided in the form of an initial guess. The first time a problem is run the unknown voltages are usually guessed to be 1.0 per unit, and the angles 0.0 radians. This is referred to as a "flat start", and it is used because it represents ideal conditions within the system, that is every bus at its nominal voltage. Subsequent runs on the same system may use the last solution as a starting point. Verification for this model will be to achieve convergence for the 118 bus test system from a flat start.

The calculation of the Jacobian results in a set of simultaneous linear equations in $\Delta\delta$ and $\frac{\Delta|E|}{|E|}$, which can be solved by any of several direct methods. When found, these angle and magnitude corrections are applied to the last values used, the real and reactive powers are recalculated for each node, and new values for ΔP and ΔQ are found. If any of these exceed a stipulated maximum mismatch value, the problem is continued. A new Jacobian is calculated and the process continues until all mismatches are below the maximum, and no voltage controlled bus is exceeding its reactive power limits. At this point convergence has been reached.

The criteria for convergence of the Newton Raphson method are two (11,12): first, as mentioned in Chapter II, the initial guess must be close enough to the final solution that the approximation made by casting off all $\Delta\delta$ and $\frac{\Delta|E|}{|E|}$ terms of order higher than one is still reasonably valid; and second, the Jacobian matrix must be non-singular. Mathematical proofs of convergence (11) are based on assumptions that the two conditions above prevail. There is no way to assure beforehand that they do in fact prevail, but if it is found that the process does not properly converge the problem must then lie in a bad initial guess, a singular Jacobian, or both. If the initial guess is too far from the final result, the process may converge to an infeasible solution, or diverge, both of which cases will be apparent in the output. The way around this problem is to move away from the "flat start" by revising the guesses for voltage magnitude and angle downward slightly for buses away from sources until a satisfactory set is found. If a computation is made using the last solution as a starting point, this difficulty is much less likely to arise. A flat start may also cause problems in the Jacobian. If, in addition to the flat start, all line admittances are identical an interesting condition results. All of the off-diagonal non-zero H and L terms will be the same: the negative of the line susceptance. The diagonal H and L terms for each node will be the negative of the sum of all the off diagonal terms, plus or minus the reactive power calculated for the node. All of the off-diagonal non-zero N terms will be the negative of all the off-diagonal non-zero J terms, and equal to the line conductance. The diagonal terms in each case are equal to the negative of the sum of all the off-diagonal terms, plus or minus the real power calculated for the node. However, since the flat start means 1.0 per unit voltage and 0.0 radians at each node, and if the swing bus designated the same way, all calculated real and reactive powers will be zero. Thus a pattern of symmetry emerges which could well lead to a singular Jacobian. The solution is simple, and is the same as the previous one: change some of the initial guesses. This will change the values of diagonal terms by contributing P and Q at effected nodes, and will also change off-diagonal terms since voltage components are considered in them. The singular Jacobian is of minor concern in the first iteration only. In the second and later iterations all of the voltages, and therefore all of the Jacobian terms will have been changed and the singular Jacobian is very unlikely to occur. Figure 7 is a flow chart of the load flow program. Appendix A contains a listing of the program dimensioned to handle the 118 bus test system. The input data and the results are also included in the appendix. The program establishes several complex quantities, and then makes the real and imaginary parts of each available separately through equivalence statements. The number of lines, number of buses, maximum permissable mismatch and maximum number of iterations allowed are read in. Next the bus data are read, including type code, voltage magnitude, voltage angle in degrees and bus load and generator data. The angles are immediately converted to radians, and the rectangular components of the voltages are calculated. The voltages are stored in both polar and rectangular form because both forms are used by the program. The line admittances are read and the bus admittance matrix is constructed. Bus

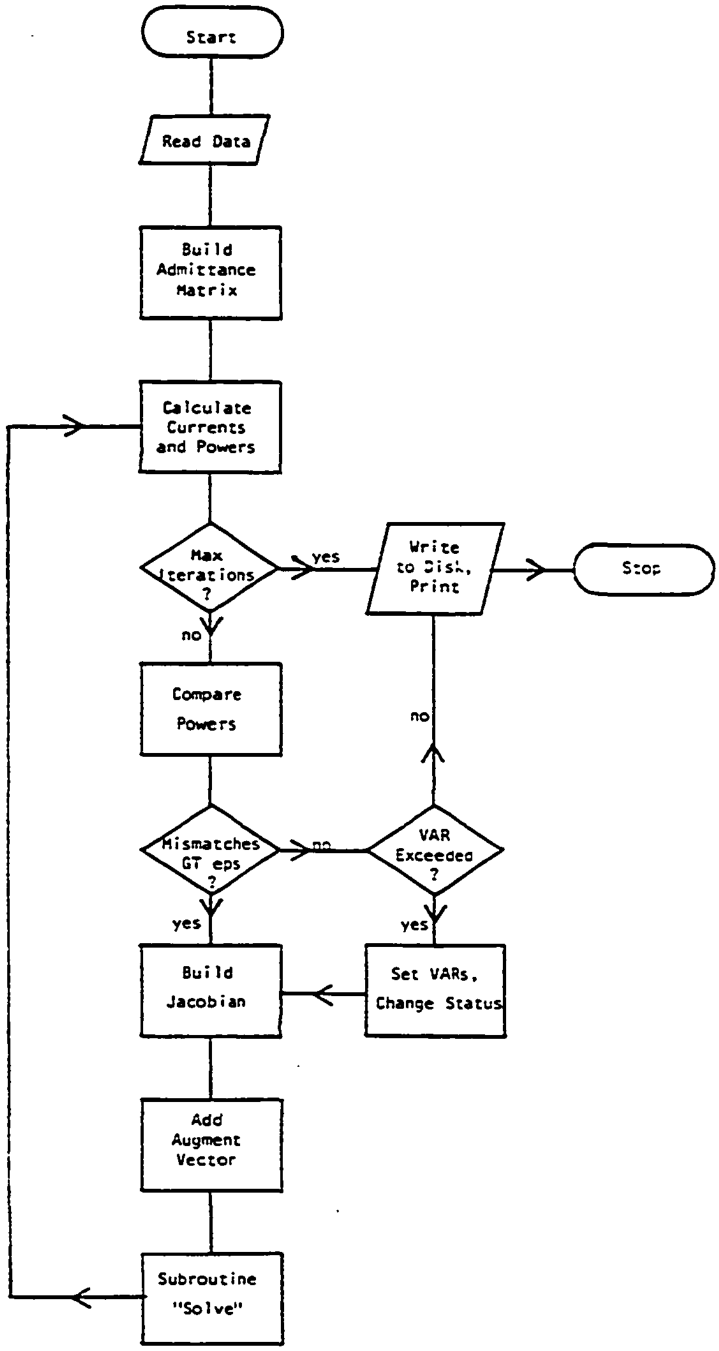


Figure 7. Flow Chart of Load Flow Program

currents and bus powers are calculated using complex quantities. The calculated real and reactive powers are subtracted from the specified powers, and the resulting ΔP's and ΔQ's are compared to the prescribed maximum mismatch. Then the Jacobian matrix is built. In this program the Jacobian is constructed with all H terms grouped at the upper left, all N terms at the upper right, all J terms at the lower left and all L terms at the lower right. There will be H terms for each bus, but the numbers of N, J and L terms will depend on the number of buses of each type, load and voltage controlled. Therefore the bus types are counted. In order to simplify the writing of the loops to actually build the Jacobian it was decided to treat all of the load buses first. This required reordering the buses so that the voltage controlled buses followed the load buses in sequence. This and the previous step are omitted for iterations past the first if no bus has changed status. The Jacobian is built, and the augment vector of ΔP's and ΔQ's is added. Subroutine "Solve" is called to solve the linear equations. It employs the "Gauss-Jordan Complete Elimination" technique, and returns the solution vector of $\Delta\delta$'s and $\frac{\Delta|E|}{|E|}$'s to the main program. The voltage magnitudes and angles are adjusted by the correction vector, and the new quantities are used to calculate new currents and powers. The new powers are subtracted from the specified values and the new ΔP's and ΔQ's are compared with the maximum mismatch. If any ΔP or

ΔQ is greater than the maximum permitted, a new Jacobian is built and the process goes through another iteration. If all ΔP's and ΔQ's are less than the maximum permitted, the calculated reactive power for each voltage controlled bus is compared to the limits provided. If any such bus exceeds a reactive power limit, the reactive power required for that bus is then specified to be that limit, the voltage is set free to vary, and the type is changed to load bus. The Jacobian is recounted and again reordered and the process continues. If the reactive power at each voltage controlled bus is within limits, the problem is solved and the results are printed. If the program goes through the maximum number of iterations without converging, the process is stopped and the latest results are printed.

System Reduction

The "Linearized Reduction" technique described briefly in Chapter II was selected for inclusion in this thesis. The technique was tested (5) against the other methods also described in the last chapter and it was found to be superior in accuracy and convergence characteristics. The severe contingencies tested included the simultaneous outage of six lines within the reduced system. Perturbations contemplated for the present effort involve altering the load at most two buses, an event much smaller in scope. Also, the technique involves manipulating the Jacobian matrix, and routines to build the Jacobian have already been written for the load flow program.

Three types of buses are treated; buses to be reduced, buses to be retained, and buses on the boundary between the first two sets. Branches leading from the reduced portion to boundary buses are reduced. Load, generation and any other shunt element connected to a boundary bus are considered part of the retained system, as are any branches connecting two boundary buses.

Before the procedure can begin a load flow calculation must first be made on the entire system. The voltage magnitudes and angles which result from this "base case" load flow are preserved and used as input data to the reduction segment. Figure 8 is a flow chart of the process and Figure 9 is a small system which will be used as an example.

In this example, bus 1 is the swing bus and the other six are normal load buses. Buses 2 and 4 will be reduced, buses 6 and 7 will be retained, and buses 1,3, and 5 are the boundary buses. Now consider a set of mismatch equations written for the buses in the reduced system only. These equations (call them f_1) will necessarily be functions of the voltage magnitudes and angles in the reduced system (call them collectively x_1) and also of the voltage angles and magnitudes of the boundary nodes (call them collectively x_2). If Jacobian terms are calculated from the base case voltages the mismatches will be zero, and $J_1(\Delta x_1) + J_2(\Delta x_2) = 0$ (1)

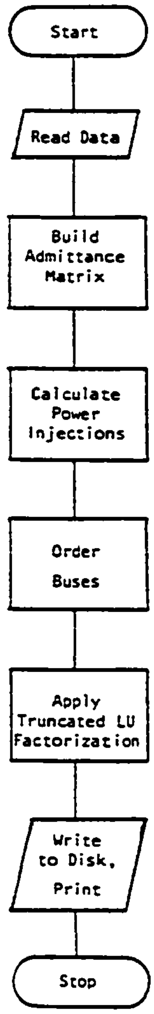


Figure 8. Flow Chart of Reduction Process.

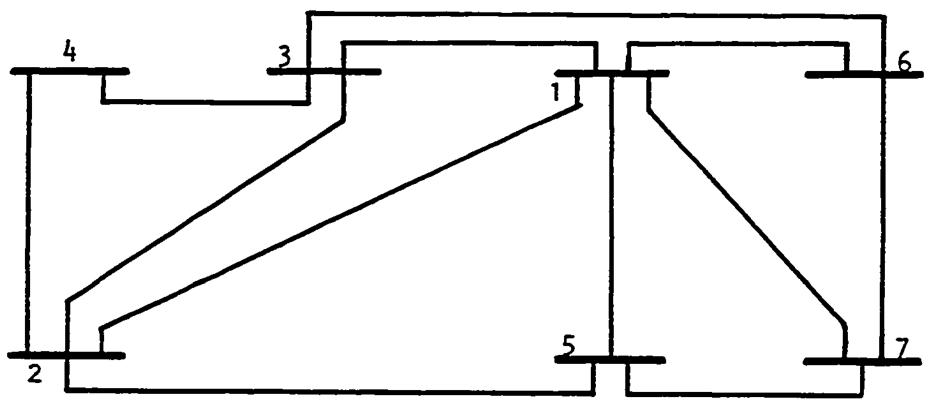


Figure 9. Example System

where $J_1 = \frac{\partial f_1}{\partial x_1}$ and $J_2 = \frac{\partial f_1}{\partial x_2}$. The mismatch at the boundary nodes will not be zero however, as the power flowing from the reduced portion to the retained portion (Sb) must appear here. This set of mismatch equations (call them f_2) will also be a function of the vectors x_1 and x_2, and $\Delta S_b = J_3(\Delta X_1) + J_4(\Delta x_2)$ or $S_b = S_{bo} + J_3(\Delta x_1) + J_4(\Delta x_2)$ (2)

where S_{bo} is a vector of base case power injections, $J_3 = \frac{\partial f_2}{\partial x_1}$ and $J_4 = \frac{\partial f_2}{\partial x_2}$.

The elements J_1, J_2, J_3 and J_4 are nothing more than the normal H,N,J and L terms calculated for base case voltage conditions, with the buses ordered so that the reduced buses are treated first and the retained buses are omitted.

When appropriate admittance terms and base case voltages and angles are applied to the example system, the Jacobian terms calculated for the reduced part, including the boundary buses are:

34.68135	-16.02254	12.24059	-4.71933	-5.30723	-5.30078	-1.54604	-1.51320
-15.64964	19.51630	-5.83207	5.89099	0.0	-3.86665	0.0	-1.25299
-11.04125	4.71933	34.98100	-16.02254	1.54604	1.51320	-5.30723	-5.30078
5.83207	-7.09107	-15.64964	19.31560	0.0	1.25299	0.0	-3.86665
-5.17339	0.0	-1.94757	0.0	51.18033	-30.68173	16.66301	-10.04373
-5.14852	-3.84511	-1.96997	-1.31760	-30.57162	44.63440	-10.37407	14.39696
1.94757	0.0	-5.17339	0.0	-17.46161	10.04373	51.19348	-30.68173
1.96997	1.31760	-5.14852	-3.84511	10.37407	-15.29720	-30.57162	44.44762

Note that the matrix is 8 x 8. Bus 1, the swing bus, contributes no terms, and the load buses 2,3,4 and 5 contribute two each. The terms pertaining to the boundary buses, 3 and 5, are located in the lower right corner.

Solving Equation (1) for Δx_1 and substituting into Equation (2) yields

$$S_b = S_{bo} + (J_4 - J_3(J_1)^{-1}J_2)\Delta x_2.$$

Define $J_{cor} = J_4 - J_3(J_1)^{-1}J_2$;

then

$$S_b = S_{bo} + J_{cor}(\Delta x_2) \tag{3}$$

Straightforward calculation of J_{cor} using the four matrices would be a formidable task. Fortunately that labor is not necessary, since the elements of J_{cor} are precisely equal to items found in the spaces formerly occupied by J_4 when the matrix $\begin{bmatrix} J_1 & J_2 \\ J_3 & J_4 \end{bmatrix}$

is subjected to a "Lower-upper factorization" process which is truncated as it reaches J_4. The LU factorization algorithm is fast and easy to program. When the truncated LU factorization is applied to the Jacobian matrix of the sample system, the result is

49.92146	-32.69270	16.25859	-10.70158
-32.57552	40.66925	-11.05359	13.01385
-17.03319	10.69329	49.93910	-32.69312
11.04526	-14.04947	-32.57574	40.46370

This is J_{cor}. The base case power injections at the boundary buses (S_{bo}) are also calculated, and are found to be

Bus Number	Real Power	Reactive Power	Voltage Magnitude	Voltage Angle
1	0.2413	0.0547	1.0118	-0.1016
3	0.3181	0.0854	1.0090	-0.1070

Now write a set of mismatch equations f_3 to represent the voltage magnitudes and angles (x_3) at the retained buses, including the boundary buses, ignoring the effects of the reduced part. Then when the boundary injections S_b are considered, $S_b + f_3(x_3) = 0$

$$\text{or } S_{bo} + J_{cor}\,(\Delta x_2) + f_3(x_3) = 0 \qquad (4)$$

Calculate the standard Jacobian for f_3. Call it J_5, and Equation (4) can readily be solved by Newton's method. The total Jacobian for the reduced part is $J_6 = J_{cor} + J_5$. (5) This is not a simple matrix addition however, as the two matrices are of different dimensions. The elements of J_{cor} must carefully be added only to those elements of J_5 which apply to boundary buses.

In the example system, the Jacobian terms are found for the retained part, including the boundary buses. This includes buses 1,3,5,6 and 7. Once again, the swing bus, number 1 contributes no terms. The elements of J_{cor} are added to the elements of the Jacobian which pertain to buses 3 and 5.

Equations 3 and 5 have profound implications. Equation 3, $S_b = S_{bo} + J_{cor} (\Delta x_2)$, states that the power mismatch is corrected at boundary nodes only, by a constant amount plus another amount linearly proportional to the voltage magnitude and angle deviations from base case conditions at those nodes. That is why the voltage magnitudes and angles were included with injected power data for the example problem. Equation 5, $J_6 = J_{cor} + J_5$, states that the Jacobian for the retained network can be found by first calculating the Jacobian for the retained nodes alone, and then adding a constant correction factor to those terms deriving from boundary nodes.

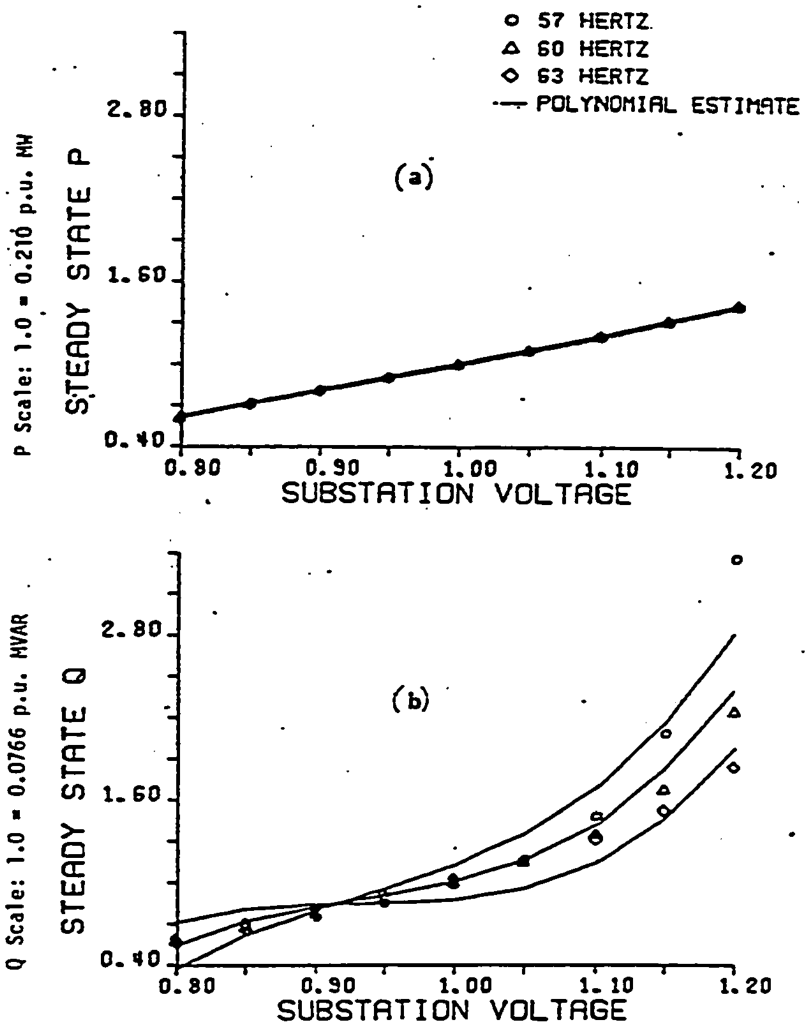
Appendix B contains a listing of the computer program written to accomplish this reduction. Input data and results for the 118 bus system are included. The program reads the number of branches, the number of buses, and then the bus type, base case voltage magnitude and base case voltage angle for each bus. It then reads the admittance of each branch and builds the bus admittance matrix. The power injected at each boundary node is calculated. The buses are reordered as follows: load bus to be reduced, voltage controlled bus to be reduced, load bus at the boundary, and voltage controlled bus at the boundary. Buses to be retained are discarded, and the Jacobian matrix is constructed. Lower-upper factorization is applied to the Jacobian and truncated at the boundary buses. The elements in the resulting matrix which were originally derived for the boundary buses are written to a direct storage device, along with the list of power injections and base case voltages and angles for the boundary buses.

Appendix B also contains a listing of the load flow program modified to handle the reduced system. Changes were made to read in the data generated by the reduction program; calculate Δx_2, the deviation at the boundary nodes from base case conditions; apply the corrections S_b to the mismatches calculated for the boundary nodes; and to add the elements of J_{cor} to the Jacobian of the reduced system. A list of output for the reduced system is included in Chapter V.

Load Representation

The University of Texas at Arlington, under EPRI contract, has conducted extensive testing of the behavior of electrical loads under varying conditions of voltage and frequency. Several item in common use were laboratory tested to find their real and reactive power at voltages ranging from 65% to 135% of rated, at frequencies of 57, 60 and 63 Hz. The measured values were used to produce equations in ΔV and ΔF for both real and reactive power. Since load flow problems, and this thesis, concern themselves with steady state operation, all terms relating to ΔF will be ignored ($\Delta F = 0$), and no further reference will be made to frequency. The different loads were then grouped according to how they would appear in a typical application, and equations were then derived to represent that application, such as Residential Summer South, Residential Winter South, Commercial Summer South, etc. These equations are all of the form $P = P_o + P_1 \Delta V + P_2 \Delta V^2 + \ldots$ where P is the real power drawn by the load, the P_i are coefficients and ΔV is the difference between the bus voltage and the device rated voltage, in per unit. Similar equations were produced for Q. Figure 10 is an extract from a preliminary report on the UTA work. It shows the points plotted, the curve fitted, and the coefficients found for a typical residential summer south load.

Appendix C contains a listing of the load flow program written to include this load representation and the reduced system. Four different load types are treated; general, industrial, commercial and residential. The coefficients for the four types are read, and codes are given to indicate which type to use at a particular node on the distribution system. In the general load, P_o is unity and the other P_i are zero; this is applied to a load where particular information is



$$P = 1.0 - 0.0670\ \Delta F + 1.96\ \Delta V + 0.501\ (\Delta V)^2 + 1.77\ (\Delta V)^3$$
$$Q = 1.0 - 2.35\ \Delta F + 2.40\ \Delta V + 11.6\ (\Delta V)^2 + 55.6\ (\Delta V)^3 - 29.2(\Delta F)(\Delta V)$$

Figure 10. Residential Summer South Load Model

not available. The algorithm uses the latest voltage calculated for a node, finds ΔV, applies the equations, and then uses the resulting P and Q to find the ΔP and ΔQ for the node. Input and output lists are also provided.

CHAPTER V

INPUT REQUIREMENTS AND SEGMENT INTERPLAY

The complete model involves three separate segments: base case load flow; system reduction and reduced system load flow; system reduction and reduced system load flow with distribution added and including voltage variable loads. In this chapter the input required for each will be described, and the manner in which the outputs from the first two segments are used by following segments will be shown.

Base Case Load Flow

The base case load flow segment is run first, to determine conditions within the system with nominal voltages and loads attached and without contingencies or perturbations. It includes the entire transmission system; in this example the IEEE 118 bus test system. The solution to the base case load flow is used as a starting point for the system reduction algorithm. The input to the base case load flow is read from a single disk file, a copy of which is provided in Appendix A. It includes one line of parameters, specifying the number of lines and buses, and the maximum mismatch and number of iterations to be permitted. Then follow 118 lines of bus data, giving the initial voltage magnitude and angle, and all load and generation attached, for each bus. The last section contains 179 lines of line data, giving the end points and admittance, for each line.

The base case load flow output is made up of two parts. (See Appendix A) The first part is a listing of iteration numbers and bus numbers showing if any bus had its status changed from voltage controlled bus to load bus because of a demand for reactive power beyond its limit. The listing also shows the value set for the reactive power at any such bus. The second part includes column headings and gives the iteration at which satisfactory convergence was achieved, followed by a listing by bus number, of bus type, bus voltage magnitude and angle, real and reactive power calculated, and any discrepancy (mismatch) between the calculated and required powers. The second part, less the iteration number and column headings is written to a disk file for use by the next segment.

System Reduction

The input data for the system reduction segment is read from two separate disk files. The first file is the same as the input file for the base case load flow. Not all of the data are needed however. From the first line only the first two items are read, i.e. the number of lines and the number of buses. From the 118 lines of bus data only the shunt susceptance is read, as this contributes to the admittance matrix. All of the information is needed from the 179 lines of admittance data.

The second input data file is the file built by the output from the base case load flow, with a second column of bus codes added, to indicate which buses are to be reduced, which are to be retained, and which are boundary buses. (Appendix B)

The output (see Appendix B) first echos the input bus numbers, types, voltage magnitudes and angles and shunt susceptances. Then there follows the Jacobian correction matrix, a square matrix of numbers, in this case the dimensions are 9 x 9. The final portion of output lists for each boundary bus the bus number, the real and reactive power injected at that bus, and the base case voltage magnitude and angle (in radians) for the bus. The last two parts, the correction matrix and the boundary bus data, are written to a disk file for use by the final segment.

Distribution Load Flow

In this final segment the load flow problem is solved for the reduced transmission system, with a distribution system attached to one of the retained buses in place of the original load there, and with the voltage variable condition considered for the loads. The input is read from two disk files. The first file (see Appendix C) is quite similar to the input file to the base case load flow. The first line contains exactly the same information, i.e., number of lines, number of buses, maximum mismatch and maximum iterations. The next four lines are coefficients used to describe the voltage variability of the loads. The bus data which follows are the same as that provided for previous segment except that a third bus type code is added to indicate the type of load attached.

The second input file is the one which was written to the disk by the previous segment: the Jacobian correction matrix and the list of power injections and voltages at the boundary buses.

The output from the segment is the solution to the combined load flow problem. It is in the same form as the second part of the output from the first segment. It gives the iteration number when convergence was reached and the voltages, angles, calculated real and reactive powers and mismatches for each bus. An example is provided as the last item in Appendix C.

CHAPTER VI

TEST RESULTS

During the development of the three segments of this load flow model many intermediate tests were made, mostly using small sample systems, to check and verify proper behavior in the various processes. With one exception these developmental tests will not be mentioned again. This chapter will report on the performance tests using the IEEE 118 bus test system and two different distribution systems as promised in Chapter II. The exception mentioned above will be a brief presentation of an impressive test of the validity of the reduction technique.

118 Bus Test System

The IEEE 118 bus test system is shown on three pages as Figure 11 . The buses are numbered. The arrows on the buses indicate loads. A large circled "G" represents a generator, and a similar "C" represents a synchronous condensor. The small plus signs (+) found near some buses (see bus 24) indicate that those buses appear on more than one page and are therefore connecting points between the pages.

Other than their basis in the 118 bus test system, the two tests run had nothing in common. They used different

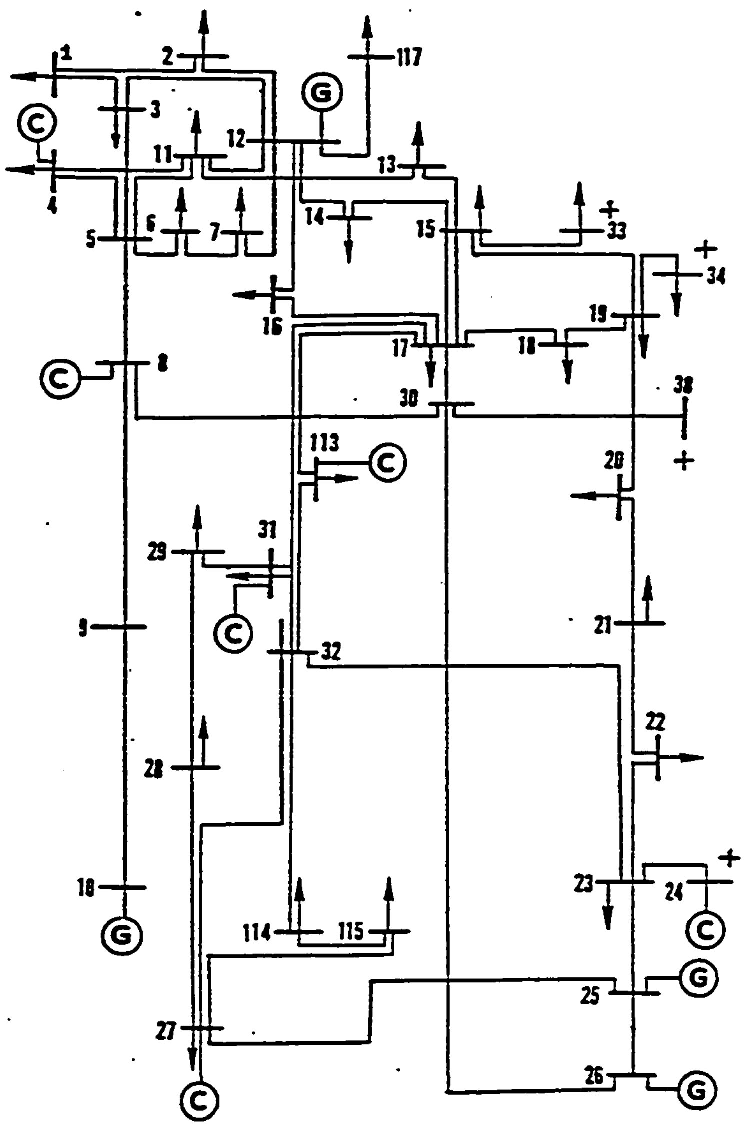


Figure 11. IEEE 118 bus Test System

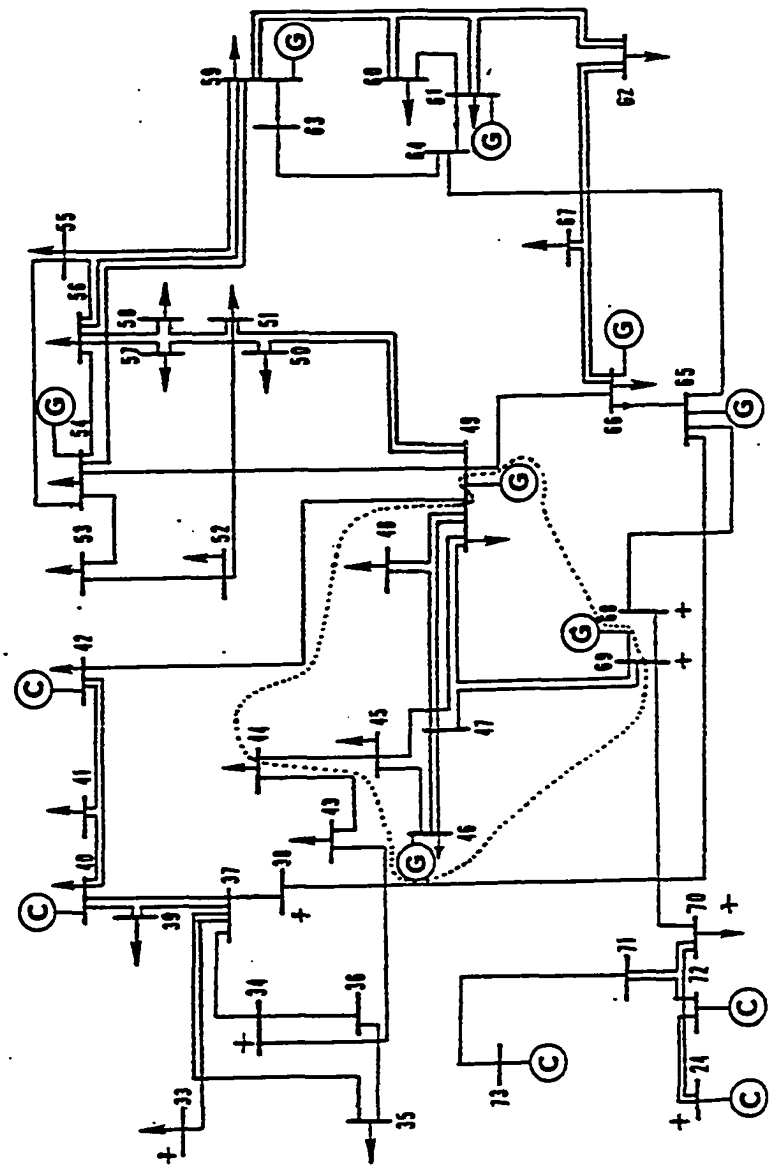


Figure 11. IEEE 118 bus Test System

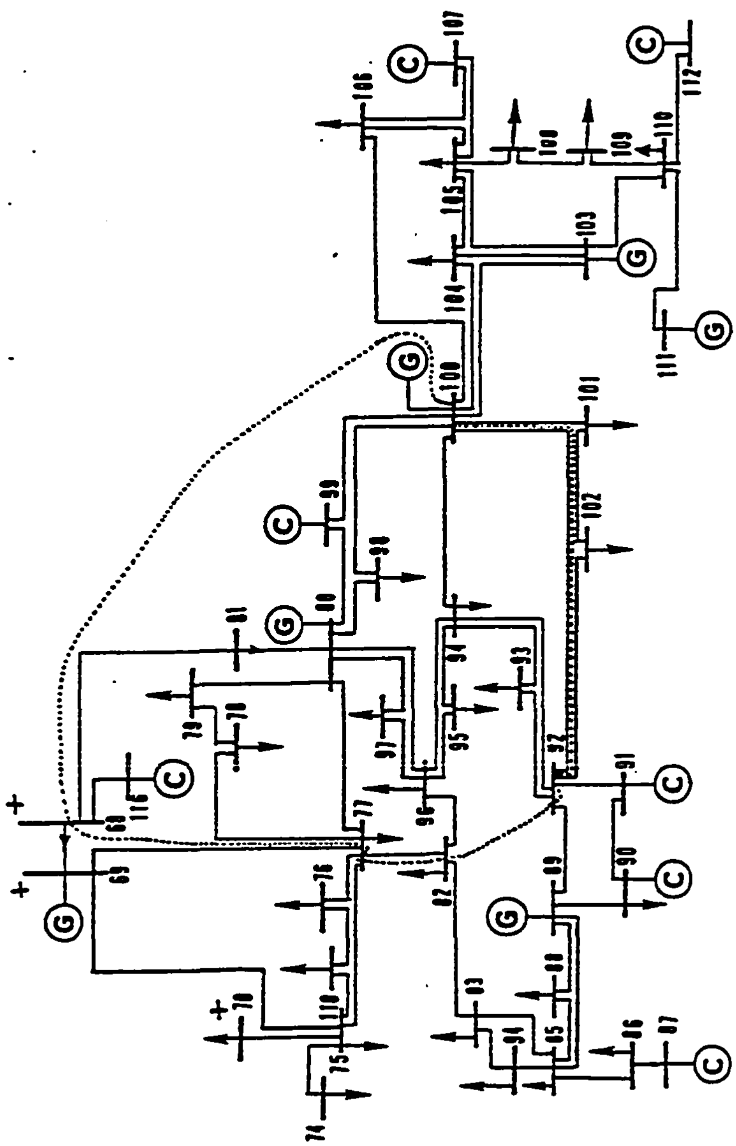


Figure 11. IEEE 118 bus Test System

portions of the 118 bus system, and attached different distribution systems to different buses in the system. The two tests will be described here concurrently, with differences being pointed out as they are encountered. The concurrent description is not meant to imply that the tests were actually run concurrently. In reality, the first test was completely finished and successful before the second was begun.

The second test was felt to be the more significant of the two, and it was used to generate the input and output examples of Appendices A thru C. The reasons for the choice will be given later in this section.

The first base case load flow was run using bus 69 as the slack bus. That bus was used because the original data set from which the system parameters were first obtained had specified it as the slack bus. The second base case load flow used bus 80 as the slack bus.

The Reduced System

A load flow problem involving a reduced system is still a load flow problem, and it must therefore employ a slack bus. Moreover, if the results using the reduced system are expected to be the same as for the complete system, the same bus must be used as slack bus in both cases. The portions of the 118 bus system which were preserved in the system reductions are outlined by the dotted lines on pages 2 and 3 of figure 11 . The buses cut by the dotted lines are boundary buses.

The buses inside are retained, and all others are eliminated. In the first case buses 44, 49 and 69 are boundary buses while buses 45, 46, 47 and 48 are retained. Note that bus 69, the slack bus is also a boundary bus. In the second case buses 68, 77, 82, 92 and 100 are the boundary buses and buses 78, 79, 80, 81, 93, 94, 95, 96, 97, 98, 99 and 116 are retained. Here the slack bus, number 80 is interior to the retained portion. The second reduction is more complicated and therefore more significant in its accomplishment than the first, even more than is first apparent. The manipulations involved in a load flow problem using a reduced system of this type are all focused on the boundary buses. The Jacobian correction matrix adjusts those terms of the system Jacobian that are associated with the boundary buses, and also the powers injected at those same buses. From the discussion in Chapter II it is recalled that a load bus has two equations, a voltage controlled bus has one equation, and a slack bus has none. Therefore each load bus on the boundary contributes two to the dimension of the Jacobian correction matrix, each voltage controlled bus contributes one, and the slack bus contributes nothing. It is seen then that bus 44, a load bus makes two, bus 49, a non-converted voltage controlled bus adds one, and the slack bus, 69, adds none, and the Jacobian correction matrix is only 3 x 3. In the second case four of the boundary buses are load buses (68, 77, 82 and 92) and the fifth is voltage controlled, so the

Jacobian matrix is 9 x 9. The latter is felt to be a much more meaningful exercise. This portion of the system, and slack bus 80, were selected to provide this added complexity and therefore a more exacting test.

The intermediate test to verify the effectiveness of the reduction technique will now be presented. Several pages of input and output are provided in Appendix A, which pertain to the base case load flow, with bus 80 as slack bus. If the reduction technique is a good one, then a load flow problem run using only the reduced portion should yield the same results as one using the entire system. This was tried. All of the data having to do with eliminated buses were removed from the input data set, leaving only that shown in Figure 12.

The program was modified to read the Jacobian correction matrix and the boundary bus conditions, and to adjust the system Jacobian and calculated powers accordingly. The Jacobian correction matrix and boundary bus conditions used were those listed as output in Appendix B. The results of the test are shown in Figure 13 . When the voltages, angles, powers and mismatches here are compared with those for corresponding buses in Appendix A, it is seen that only very small differences occur, which leads to some degree of confidence in the reduction algorithm.

```
 24    17                 1.0E-03   20
 68 1      1.000    0.0     0.0     0.0     0.0     0.0    0.0     0.0     0.0
 77 1      1.000    0.0     0.610   0.280   0.0     0.0   -0.200   0.700   0.0
 82 1      1.000    0.0     0.540   0.270   0.0     0.0    0.0     0.0     0.200
 92 1      1.000    0.0     0.650   0.100   0.0     0.0   -0.030   0.090   0.0
100 1 1    1.030    0.0     0.370   0.180   2.520   0.0   -0.500   1.550   0.0
 78 2      1.000    0.0     0.710   0.260   0.0     0.0    0.0     0.0     0.0
 79 2      1.000    0.0     0.390   0.320   0.0     0.0    0.0     0.0     0.200
 80 2 2    1.035    0.0     1.300   0.260   4.770   0.0   -1.650   2.800   0.0
 81 2      1.000    0.0     0.0     0.0     0.0     0.0    0.0     0.0     0.0
 93 2      1.000    0.0     0.120   0.070   0.0     0.0    0.0     0.0     0.0
 94 2      1.000    0.0     0.300   0.160   0.0     0.0    0.0     0.0     0.0
 95 2      1.000    0.0     0.420   0.310   0.0     0.0    0.0     0.0     0.0
 96 2      1.000    0.0     0.380   0.150   0.0     0.0    0.0     0.0     0.0
 97 2      1.000    0.0     0.150   0.090   0.0     0.0    0.0     0.0     0.0
 98 2      1.000    0.0     0.340   0.080   0.0     0.0    0.0     0.0     0.0
 99 2 1    1.010    0.0     0.0     0.0    -0.420   0.0   -1.000   1.000   0.0
116 2 1    1.005    0.0     0.0     0.0    -1.840   0.0  -10.000  10.000   0.0
  68    81    4.13696 -49.15681    0.80800
  68   116   18.64508-248.60162    0.16400
  77    78   22.09615 -74.05199    0.01260
  77    80    8.90905 -27.30460    0.07000
  77    82    3.65011 -10.44814    0.08180
  78    79    8.64664 -39.07002    0.00640
  79    80    3.00028 -13.53972    0.01860
  80    96    1.03514  -5.29203    0.04940
  80    97    2.02021 -10.31082    0.02540
  80    98    1.94596  -8.83043    0.02860
  80    99    1.02029  -4.62951    0.05460
  81    80    0.0     -27.02702    0.0
  82    96    5.27440 -17.25575    0.05440
  92    93    3.28383 -10.79337    0.02180
  92    94    1.76335  -5.79230    0.04060
  92   100    0.71034  -3.22380    0.07720
  93    94    3.80837 -12.50101    0.01880
  94    95    6.41462 -21.09048    0.01100
  94    96    3.25067 -10.50124    0.02300
  94   100    4.83585 -15.75728    0.06040
  95    96    5.20627 -16.65398    0.01480
  96    97    2.12752 -10.88355    0.02400
  98   100    1.18095  -5.32467    0.04760
  99   100    2.59602 -11.72536    0.02160
```

Figure 12. Reduced System Test Data

```
ITERATION NUMBER   3
  BUS      BUS     BUS     BUS          POWER              MISMATCH
 NUMBER   TYPES  VOLTAGE  ANGLE     REAL    REACTIVE    REAL     REACTIVE
   68     1  0   1.0167  -1.041  1.49024  1.76197   0.00003   0.00021
   77     1  0   1.0032  -2.432 -1.09506 -0.36194  -0.00000  -0.00000
   82     1  0   0.9986  -2.079 -0.07208 -0.34398   0.00001   0.00006
   92     1  0   1.0120   4.078  1.39187 -0.30596  -0.00000   0.00002
  100     1  1   1.0300  -1.296 -0.06879  0.72127  -0.00001   0.0
   78     2  0   1.0004  -2.716 -0.71001 -0.26005   0.00001   0.00005
   79     2  0   1.0056  -2.375 -0.39000 -0.31999  -0.00000  -0.00001
   80     2  2   1.0350   0.0    0.0      0.0       0.0       0.0
   81     2  0   1.0290  -0.704  0.00000 -0.00005  -0.00000   0.00005
   93     2  0   1.0038   1.282 -0.12000 -0.06999  -0.00000  -0.00001
   94     2  0   1.0036  -0.716 -0.30003 -0.16008   0.00003   0.00008
   95     2  0   0.9921  -1.617 -0.41999 -0.30998  -0.00001  -0.00002
   96     2  0   1.0009  -1.721 -0.37999 -0.14997  -0.00001  -0.00003
   97     2  0   1.0131  -1.216 -0.15000 -0.09001   0.00000   0.00001
   98     2  0   1.0253  -1.693 -0.34000 -0.08001   0.00000   0.00001
   99     2  1   1.0100  -2.081 -0.42000 -0.31259   0.00000   0.0
  116     2  1   1.0050  -1.406 -1.83998 -2.88768  -0.00002   0.0
```

Figure 13. Reduced System Test Output

The Distribution Systems

Data sets describing distribution systems in two different small cities in Oklahoma were obtained from a local engineering consulting firm. In both tests a distribution system was attached to a bus interior to the retained portion of the 118 bus system, whose original assigned load was very nearly that of the total load on the respective distribution system. In the first test the distribution system of Fairview, Oklahoma, made up of two feeders, was attached at bus 48, and the load at bus 48 was reduced. The problem converged in five iterations, and the results are shown in Appendix D.

The second test used eight feeders from a single substation in Altus, Oklahoma. This second test was again considered to be more meaningful for the reason that the data used was more accurate, especially in the area of load description. The distribution system is shown schematically in Figure 14 . Commercial loads are indicated by a "C", and industrial loads by an "I". All other loads are residential. The figure is not drawn to scale, nor is it geographically correct. It is valid for connections and load type only. The results, in Appendix C, show that the problem was solved in five iterations.

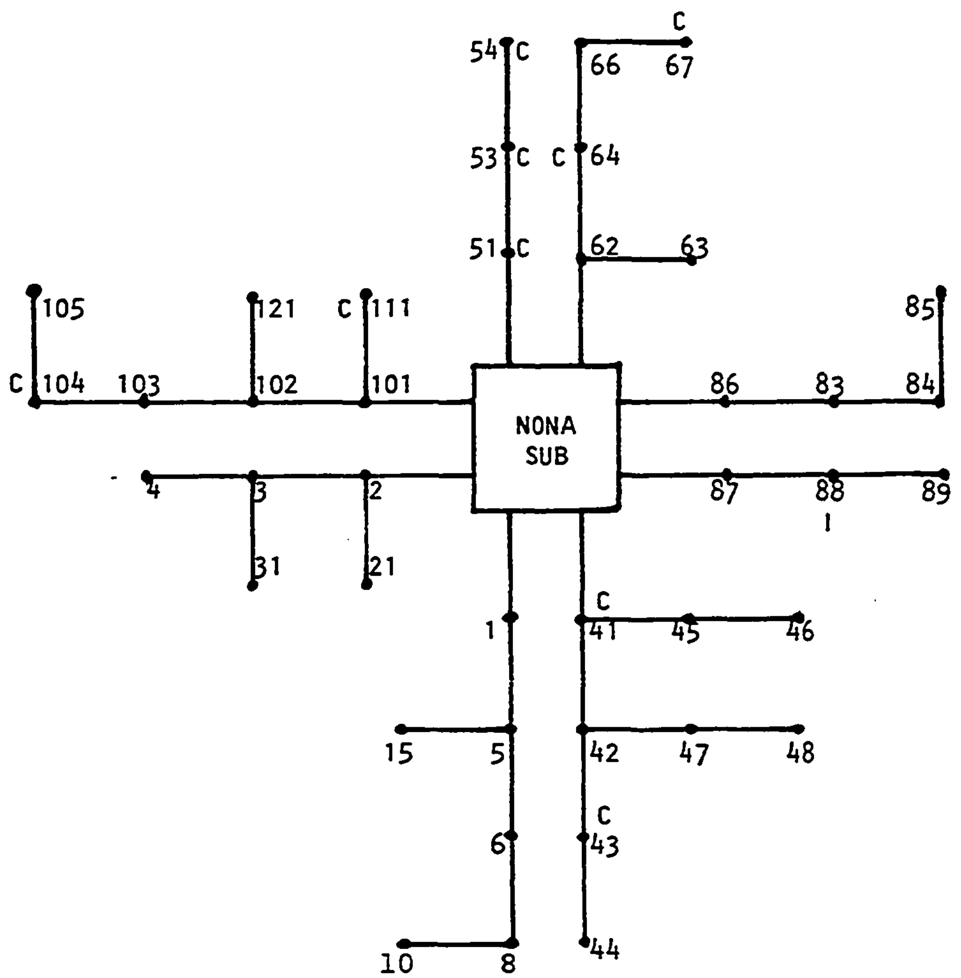


Figure 14. Distribution System, Altus, Oklahoma

Model Validation

In order to be a useful computational tool, the combined load flow model must produce correct results. A validation test was performed to compare the output of the new model with that obtained by conventional methods for the same problem. The test employed a Gauss-Seidel load flow program available at the University of Oklahoma, and a voltage profile program from Central Area Data Processing, St. Peters, Missouri.

In the first step of the test a load flow problem was run on the Gauss-Seidel program using the IEEE 118 bus test system, with bus 80 as slack bus. The program is written to use the "per unit" system throughout, just as the combined model does. The input conditions were:

Voltage: 1.0 per unit volts at load buses.
As specified at generator buses.

Bus Angle: 0 degrees at all buses.

Bus Power: Generation and load as specified.

This arrangement duplicated the system and conditions used in the test of the combined model. The result of this run was compared to output data for the base case load flow given in Appendix A. The voltage levels were found to be virtually identical. The voltage at bus 97 was noted, since it had been used as the substation bus in the combined model.

That voltage became the starting point for the voltage profile program. The program begins with real voltages, loads and impedances, and converts quantities as needed using the turns ratio of the transformers. It asks for the substation bus voltage in terms of 120 volts. The voltage found above for bus 97 (1.013 pu) was converted to a 120 volt base and the resulting 121.56 volts was entered for eight separate voltage profile problems, one for each feeder out of the substation.

The real and reactive loads found for the eight feeders were added up and the total turned out to be significantly greater than the load originally scheduled for bus 97. The load flow input data consequently was altered to reflect an increase at that bus amounting to 2000 kW and 4200 kvar. The load flow program was run again with initial conditions the same as the first run, except for the change at bus 97.

The voltage on bus 97 was found to have decreased to

1.011 per unit. That was converted to 121.32 volts and another set of voltage profiles was run.

The total real and reactive loads were found to have changed slightly from the last set, but not enough to affect the least significant digit in the input data for bus 97. The process was terminated and the latest results were compared to the model output in Appendix C. The comparison is shown in the table.

The table is divided vertically into two sections; one for the transmission portion and one for the distribution part. Bus voltages are given in per unit, in columns headed according to the source; i.e. 'Gauss-Seidel' and 'Combined' for transmission, and 'Voltage Profile' and 'Combined' for distribution. The 'Voltage Profile' also contains a figure in parentheses. That figure is the actual output of the program; the total voltage drop at the node, based on 120 volts. It was converted manually to the per unit figure for consistency. The distribution list does not include intermediate nodes, but does include the last node on each feeder and branch (See Figure 14).

The transmission figures compare very closely. The largest discrepancy is 0.002 per unit at bus 92; a boundary bus. In the distribution portion however, the differences are, in general, greater. The largest is 0.0221 per unit at bus 89.

There are at least three causes of discrepancies

	Transmission			Distribution		
Bus	Gauss-Siedel	Combined	Bus	Voltage	Profile	Combined
68	1.017	1.0167	10	0.7915	(26.3)	0.8026
77	1.003	1.0032	15	0.9625	(5.82)	0.9662
78	1.000	1.0004	4	1.0075	(0.42)	1.0071
79	1.005	1.0056	21	1.0077	(0.40)	1.0071
80	1.035	1.0350	31	1.0075	(0.42)	1.0069
81	1.029	1.0290	105	0.9959	(1.81)	0.9945
82	0.998	0.9985	111	1.0008	(1.22)	1.0005
92	1.015	1.0130	121	0.9974	(1.63)	0.9961
93	1.005	1.0037	44	0.9858	(3.03)	0.9772
94	1.004	1.0035	46	0.9916	(2.33)	0.9861
95	0.992	0.9917	48	0.9950	(1.92)	0.9916
96	1.000	1.0004	54	0.9973	(1.64)	0.9926
97	1.011	1.0112	63	0.9792	(3.82)	0.9730
98	1.025	1.0253	67	0.9719	(4.69)	0.9631
99	1.010	1.0100	85	0.9313	(9.57)	0.9280
100	1.030	1.0300	89	0.9436	(8.09)	0.9215
116	1.005	1.0050				

VOLTAGE COMPARISON TABLE

between the solutions. First, the combined model uses vector operations to solve for the actual voltage magnitude at each bus. The voltage profile program multiplies the line current by the line impedance, and uses the real part of the product as the voltage drop in the section. It then simply adds these voltage drops to obtain the total drop on a feeder. That technique would be exact only if all the bus angles were zero degrees. That is not the case, and errors are introduced.

The second reason has to do with the assignment of the loads. In the combined model the loads are assigned directly to the nodes. The voltage profile has the loads initially assigned to line sections. For the actual calculation, however, the loads are reassigned; half the load to each end node of the section. Thus only half the current for the load on a section is considered when the voltage drop is computed.

The third, and the most important difference is that the combined model uses voltage variable loads, while in the voltage profile, the loads are fixed. Even though the test was begun with identical input data, the final loads in the two problems were somewhat different, and without the same loads we can't expect the same voltages.

The comparison of the transmission results instills complete confidence in that portion of the model. Considering the three factors discussed above, the distribution data still tracks rather well with a maximum discrepancy of only 2.2%, and the differences result from the superior performance of the combined model. The model must be considered to be valid.

A second test was performed to clarify further the accuracy of the combined model. This test examined the behavior of the model as the loads on the distribution part were decreased below the original levels. Seven additional runs were made on the combined model, as the loads were varied in 10% increments, from 90% to 30% of the original. The table shows the results for five of the test runs, and for the original run. The buses included in the table are the same ones used in the previous table, which showed the validity of the model. They are the buses at the ends of the feeders and taps, and are therefore the ones of most interest in the problem. Bus 97, the substation bus is also shown.

The results are entirely consistent with expectations. Figure 15 is a graph of three representative buses, and it portrays clearly how the bus voltages increase as the loads decrease. The increases for the different increments at a given bus are similar, but not identical. This expected non-linearity is just barely discernible in the figure.

The test shows that the combined model performs well at all load levels that it would be likely to be called upon to examine.

Perusal of the table will reveal some of the utility of the model. In a distribution system, the voltage should be kept within 5% of nominal; i.e., between .95 and 1.05 per unit. Bus 10 fails that criterion for loads above 30% of the planned level. Some redesign of the feeder is definitely

	LOAD LEVEL					
BUS	100%	80%	60%	50%	40%	30%
97	1.0112	1.0128	1.0144	1.0153	1.0161	1.0169
10	0.8026	0.8474	0.8900	0.9105	0.9312	0.9523
15	0.9662	0.9767	0.9871	0.9922	0.9975	1.0028
4	1.0071	1.0095	1.0119	1.0131	1.0144	1.0156
21	1.0071	1.0095	1.0119	1.0132	1.0144	1.0156
31	1.0069	1.0093	1.0118	1.0131	1.0143	1.0156
105	0.9945	0.9995	1.0045	1.0069	1.0094	1.0119
111	1.0005	1.0043	1.0081	1.0099	1.0118	1.0137
121	0.9961	1.0008	1.0054	1.0077	1.0101	1.0124
44	0.9772	0.9855	0.9940	0.9983	1.0025	1.0067
46	0.9861	0.9926	0.9992	1.0025	1.0058	1.0092
48	0.9916	0.9971	1.0026	1.0054	1.0082	1.0109
54	0.9926	0.9981	1.0036	1.0062	1.0089	1.0116
63	0.9730	0.9821	0.9913	0.9960	1.0008	1.0054
67	0.9631	0.9741	0.9855	0.9913	0.9971	1.0028
85	0.9280	0.9454	0.9633	0.9723	0.9814	0.9907
89	0.9215	0.9397	0.9589	0.9687	0.9787	0.9887

EFFECTS OF CHANGING
INPUT LOAD LEVELS

in order. Buses 85 and 89 have problems at loads above 60%. All of the remaining buses will perform satisfactorily in their present configuration.

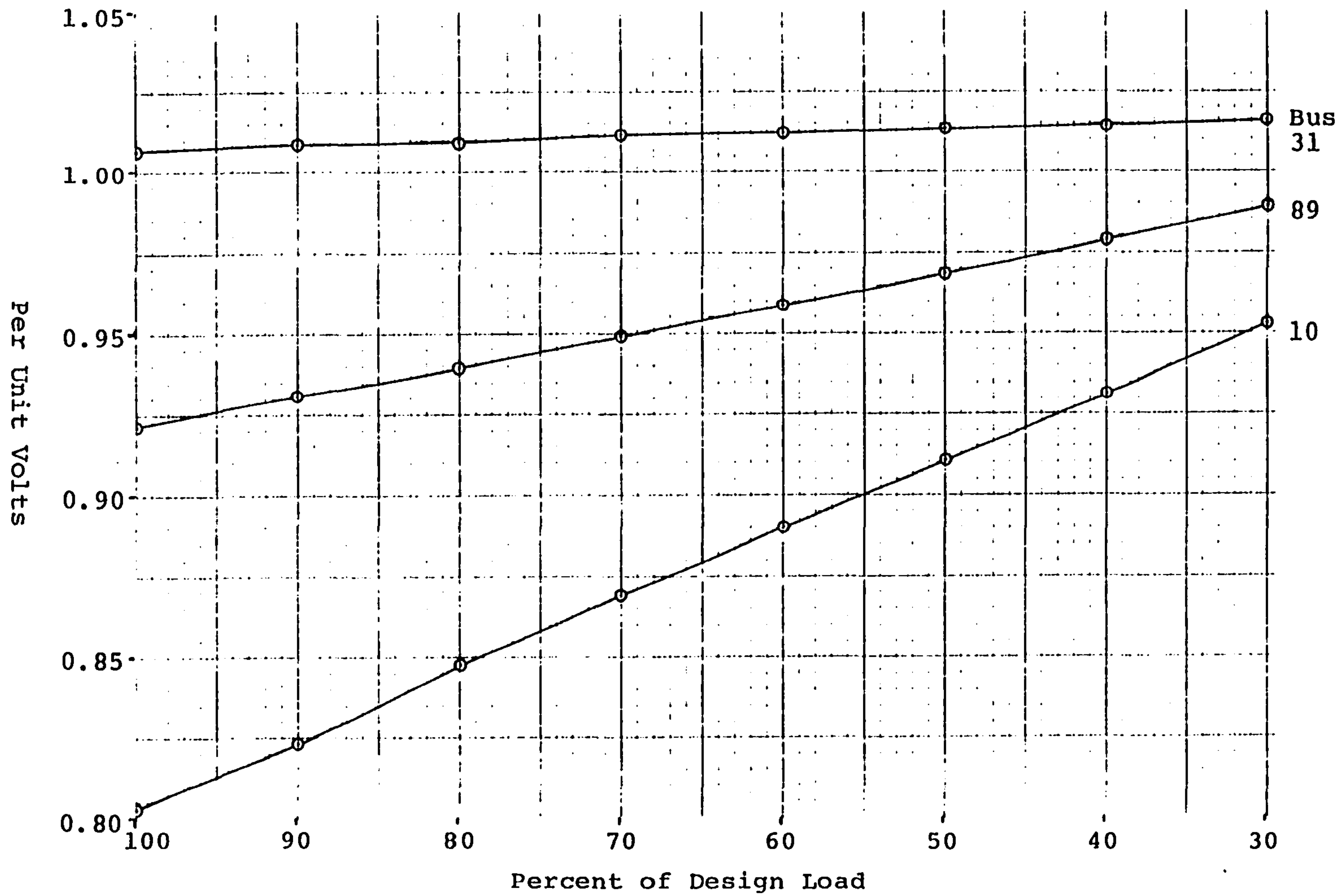


Figure 15

Voltages at Selected Buses

CHAPTER VII

CONCLUSIONS AND RECOMMENDATIONS

In the past, the transmission system load flow problem and the distribution system voltage profile problem have been addressed separately. The results of this research, as described in the previous chapter show that this separation need not be maintained; that the two problems may be treated as one. This unique approach is itself a significant contribution to knowledge in the field of power system problems, and has been the major goal of this thesis. Its attainment has been embellished by the completion of the two subtasks described in Chapter I, and in the following.

The combining of the two load flow problems was made feasible by the inclusion of the "linearized reduction" technique to eliminate from consideration large portions of the transmission system remote from the distribution system of interest. The reduction technique had been used in the past to investigate contingencies within the transmission system. The innovation in the present work; i.e., attaching a distribution system to a retained bus, considerably extends the utility of the reduction technique and is a further contribution to knowledge in this area.

The final unique feature in this research is the incorporation into the distribution system of a load representation different from that commonly used in the past. Neither the constant KVA load used in transmission load flow problems nor the constant impedance load used in distribution voltage profile problems can accurately model the behavior of real loads. The voltage-variable load representations as used in this present work are designed to pattern actual loads of different types and will therefore yield more meaningful results. This closer approximation of reality is still another contribution to knowledge afforded by this thesis.

The overall effect is the removal of the artificial barrier which has existed between the transmission and distribution load flow problems, which will ease the total task in addressing power system questions. A contingency or a load alteration on the distribution system may now be examined directly for its effect on both the transmission and the distribution systems. One need simply to change the input data and run the model, and the complete results are available. This is in sharp contrast to the previous need to analyze the systems separately, with their differing load models and solution algorithms, and the possible need for iteration back and forth as the solution in one system effected the conditions in the other. This simplification should have a considerable impact within the discipline.

Recommendations

As pointed out in Chapter IV, the thrust of this research has been to prove a concept, not to produce a production grade computer model. Now that the validity has been shown, refinements may be added to improve the utility of the model. Three relatively simple refinements are: 1) Add routines to data to per unit, so that entries may be made in raw form. 2) Provide voltage and current limits, and add routines that will flag busses and lines where those limits are exceeded. 3) Develop routines to emulate the action of voltage regulators and tap changing transformers.

The most significant refinement would be the incorporation of sparse matrix techniques (13) in the storage of the admittance and Jacobian matrices, and in the reduction algorithm used in the "solve" subroutine. This could increase the capability of the model considerably. In the admittance and the Jacobian matrices a large percentage of the elements are zero, and that percentage increases with problem size. With sparse matrix techniques only the non-zero terms are stored, thus saving large ammounts of storage space. An index file must be built to keep track of the non-zero terms, but the storage space needed by the index is far outweighed by the amount of space saved. For large systems, the capability may be improved by a factor as large as 100. (13)

REFERENCES

1. Stevenson, W.D. Jr., Elements of Power System Analysis, New York: McGraw-Hill, 1975.

2. Stagg, G.W. and El-Abiad, A.H., Computer Methods in Power System Analysis, New York: McGraw-Hill, 1968.

3. Tinney, W.F., Hart, C.E., "Power Flow Solution by Newton's Method", IEEE Transactions on PAS, November, 1967.

4. Kron, G., Diakoptics: The Piecewise Solution of Large-Scale Systems, London: MacDonald, 1963.

5. Alvarado, F.L., Elkonyaly, E.H., "Reduction in Power Systems," IEEE PES Summer Meeting, July, 1977.

6. Dimo, P., Nodal Analysis of Power Systems, Tunbridge Wells, England: Abacus Press, 1975.

7. Jones, G.A., "The Effects of System Voltage Reductions on Various Static Load Models", IEEE PES Summer Meeting, July, 1974.

8. Concordia, C., "Representation of Loads", IEEE PES Symposium on Adequacy and Philosophy of Modelling: System Dynamic Performance - 75 CHO 970-4-PWR, 1975.

9. Chen, M.S., "Determining Load Characteristics for Transient Performances", EPRI EL-849, May, 1979.

10. Van Ness, J.E., "Iteration Methods for Digital Load Flow Studies", AIEE Transactions, August, 1959.

11. Carnahan, B., Applied Numerical Methods, New York, John Wiley and Sons, 1969.

12. Jacoby, S.L.S., Iterative Methods for Non-Linear Optimization Problems, Englewood Cliffs, New Jersey, Prentice Hall, 1972.

13. Tinney, W.F., "Comments on Using Sparsity Techniques for Power System Problems", Sparse Matrix Proceedings, IBM, 1969.

14. Stott, B., "Review of Load Flow Calculation Methods", Proceedings of the IEEE, July, 1974.

15. Elgerd, O.I., Electric Energy System Theory: An Introduction, New York: McGraw-Hill, 1971

16. Stott, B., Alsac, O., "Fast De-Coupled Load Flow," IEEE PES Summer Meeting, July, 1973.

17. Sachdev, M.S., Medicherla, T.K.P., "A Second Order Load Flow Technique", IEEE PES Winter Meeting, January, 1976.

18. Iwamoto, S., Tamura, Y., "A Fast Load Flow Method Retaining Non-Linearity", IEEE Transactions on PAS, September/October, 1978.

19. Nagendra Rao, P.S., Prakasa Rao, K.S., Nanda, J., "An Exact Fast Load Flow Method Including Second Order Terms in Rectangular Coordinates", IEEE Transactions on PAS, September, 1982.

20. Sasson, A.M., "Nonlinear Programming Solutions for Load-Flow, Minimum-Loss, and Economic Dispatching Problems", IEEE Transactions on PAS, April, 1969.

21. Yu, D.C., "Optimal Load Flow Study Utilizing O. R. Techniques", Ph.D. Thesis, University of Oklahoma, 1983.

22. Nagendra Rao, P.S., Prakasa Rao, K.S., Nanda, J., "An Empirical Criterion for the Convergence of the Fast Decoupled Load Flow Method", IEEE Transactions on PAS, May, 1984.

APPENDIX A

This appendix contains the FORTRAN source listing, input data set and output list for the second test of the basic load flow segment of the program. The input is read from a single disk file, structured as follows:

The first line in the file is

179	118	1.0E-03	20
(A)	(B)	(C)	(D)

where

(A) indicates the number of lines in the system

(B) indicates the number of buses

(C) is the maximum power mismatch permitted

(D) is the maximum number of iterations allowed.

Then follow 118 lines of bus data, with all quantities in per unit, such as

31	1	0.967	0.0	0.430	0.270	0.070	0.0	-3000	3.000	0.0
(A)	(B)	(C)	(D)	(E)	(F)	(G)	(H)	(I)	(J)	(K)

where

(A) is the bus number

(B) is the bus type (0 for load bus, 1 for voltage controlled bus, 2 for slack bus)

(C) is the initial guess or specified voltage magnitude at the bus.

(D) is the initial voltage angle at the bus

(E) is the real load

(F) is the reactive load

(G) is the real power generated at the bus

(H) is the reactive power generated at the bus, if fixed

(I) is the reactive power lower limit if variable

(J) is the reactive power upper limit if variable

(K) is the value of shunt capacitance attached at the bus.

The last section of input consists of 179 lines of admittance data in per unit, as for example

15	17	6.33419	-20.97000	0.04440
(A)	(B)	(C)	(D)	(E)

where

(A) is the "from" bus number

(B) is the "to" bus number

(C) is the conductance of the line

(D) is the susceptance of the line

(E) is an added susceptance to account for line changing current.

The output list is self-explanatory.

```
C
C     NEWT118U.FORT. NOV 16. '79.
C     120 BUS SYSTEM. SMALL SOLVE ROUTINE. NO "X" VECTOR.
C     WRITES SOLUTION ON TUBE 10.
C
      COMPLEX WYE(120,120),V(120),CUR(120),P(120),S(120),CONJG,WHY
      DIMENSION Y(2,120,120),E(2,120),AYE(2,120),PWR(2,120),PQ(2,120)
      DIMENSION DP(120),DQ(120),ITYPE(120),VEE(120),ANG(120),YSH(120)
      DIMENSION GMW(120),GMV(120),GMVMIN(120),GMVMAX(120),JK(240)
      EQUIVALENCE (WYE(1,1),Y(1,1,1)),(V(1),E(1,1)),(CUR(1),AYE(1,1))
      EQUIVALENCE (P(1),PWR(1,1)),(S(1),PQ(1,1))
      REAL JAKE(240,241),LMW(120),LMV(120)
      DATA Y/28800*0.0/
C
C     READ PARAMETERS AND INITIALIZE
C
      READ(5,202)NADM,NBUS,EPS,ITMAX
      ITER=0
      NH=NBUS-1
C
C     READ BUS DATA
C
      DO 35 I=1,NBUS
      READ (5,201) IBUS,ITYPE(I),VEE(I),ANG(I),LMW(I),LMV(I),
     1GMW(I),GMV(I),GMVMIN(I),GMVMAX(I),YSH(I)
      ANG(I)=ANG(I)*3.14159/180.0
      E(1,I)=VEE(I)*COS(ANG(I))
      E(2,I)=VEE(I)*SIN(ANG(I))
      PQ(1,I)=GMW(I)-LMW(I)
      PQ(2,I)=GMV(I)-LMV(I)
   35 Y(2,I,I)=Y(2,I,I)+YSH(I)
C
C     BUILD ADMITTANCE MATRIX
C
      DO 20 I=1,NADM
      READ (5,200) IFR,ITO,WHY,CHG
      WYE(IFR,ITO)=-WHY
      WYE(ITO,IFR)=-WHY
      Y(2,IFR,IFR)=Y(2,IFR,IFR)+(CHG/2.0)
   20 Y(2,ITO,ITO)=Y(2,ITO,ITO)+(CHG/2.0)
      DO 30 I=1,NBUS
      DO 30 J=1,NBUS
   30 IF(I.NE.J) WYE(I,I)=WYE(I,I)-WYE(I,J)
      GO TO 44
C
C     CORRECT VOLTAGE ANGLE AND MAGNITUDE VECTORS
C
   40 DO 41 I=1,NH
      K=JK(I)
   41 ANG(K)=ANG(K)+JAKE(I,NJP)
      DO 42 I=NBUS,NJAC
      K=JK(I-NH)
   42 VEE(K)=VEE(K)*(1.0+JAKE(I,NJP))
      DO 43 I=1,NBUS
      E(1,I)=VEE(I)*COS(ANG(I))
      E(2,I)=VEE(I)*SIN(ANG(I))
      PQ(1,I)=GMW(I)-LMW(I)
   43 PQ(2,I)=GMV(I)-LMV(I)
```

Source Listing - Load Flow

```
C
C     CALCULATE CURRENTS AND POWERS
C
   44 DO 45 I=1,NBUS
      DP(I)=0.0
      DQ(I)=0.0
   45 CUR(I)=0.0
      ICHK=0
      DO 50 I=1,NBUS
      IF (ITYPE(I).EQ.2) GO TO 50
      DO 52 J=1,NBUS
   52 CUR(I)=CUR(I)+WYE(I,J)*V(J)
      P(I)=V(I)*CONJG(CUR(I))
      DP(I)=PQ(1,I)-PWR(1,I)
      IF (ABS(DP(I)).GT.EPS) ICHK=1
      IF (ITYPE(I).NE.0) GO TO 50
      DQ(I)=PQ(2,I)-PWR(2,I)
      IF (ABS(DQ(I)).GT.EPS) ICHK=1
   50 CONTINUE
      IF (ITER.GE.ITMAX) GO TO 110
      IF(ICHK.EQ.1.AND.ITER.GT.0) GO TO 75
      IF(ICHK.EQ.1) GO TO 54
      DO 54 I=1,NBUS
      IF (ITYPE(I).EQ.2) GO TO 54
      GVAR=PWR(2,I)+LMV(I)
      IF(ITYPE(I).EQ.1.AND.GVAR.LT.GMVMIN(I)) GO TO 56
      IF(ITYPE(I).EQ.1.AND.GVAR.GT.GMVMAX(I)) GO TO 58
      GO TO 54
   56 GMV(I)=GMVMIN(I)
      GO TO 59
   58 GMV(I)=GMVMAX(I)
   59 ITYPE(I)=0
      ICHK=1
      PQ(2,I)=GMV(I)-LMV(I)
      DQ(I)=PQ(2,I)-PWR(2,I)
      WRITE (6,205) ITER,I,GMV(I)
   54 CONTINUE
      IF (ICHK.EQ.0) GO TO 110
C
C     COUNT JACOBIAN
C
      NLD=0
      NVC=0
      DO 60 I=1,NBUS
      J=ITYPE(I)-1
      IF (J) 61,62,60
   61 NLD=NLD+1
      GO TO 60
   62 NVC=NVC+1
   60 CONTINUE
      NJAC=2*NLD+NVC
      NJP=NJAC+1
C
C     REORDER BUSSES
C
      K=0
      L=NLD
      DO 70 I=1,NBUS
```

Source Listing - Load Flow

```
      J=ITYPE(I)-1
      IF (J) 71,72,70
   71 K=K+1
      JK(K)=I
      GO TO 70
   72 L=L+1
      JK(L)=I
   70 CONTINUE
   75 ITER=ITER+1
C
C     BUILD JACOBIAN
C
      DO 80 I=1,NH
      K=JK(I)
      DO 80 J=1,NH
      M=JK(J)
      IF (K.EQ.M) GO TO 85
      A=E(1,M)*Y(1,K,M)-E(2,M)*Y(2,K,M)
      B=E(2,M)*Y(1,K,M)+E(1,M)*Y(2,K,M)
      JAKE(I,J)=(A*E(2,K))-(B*E(1,K))
      IF (I.LE.NLD.AND.J.LE.NLD) JAKE(I+NH,J+NH)=JAKE(I,J)
      GO TO 80
   85 JAKE(I,J)=-PWR(2,K)-(Y(2,K,K)*VEE(K)*VEE(K))
      IF (I.LE.NLD.AND.J.LE.NLD) JAKE(I+NH,J+NH)=JAKE(I,J)+2.*PWR(2,K)
   80 CONTINUE
      DO 90 I=1,NH
      K=JK(I)
      DO 90 J=NBUS,NJAC
      M=JK(J-NH)
      IF (K.EQ.M) GO TO 95
      A=E(1,M)*Y(1,K,M)-E(2,M)*Y(2,K,M)
      B=E(2,M)*Y(1,K,M)+E(1,M)*Y(2,K,M)
      JAKE(I,J)=(A*E(1,K))+(B*E(2,K))
      JAKE(I+NH,J-NH)=-JAKE(I,J)
      GO TO 90
   95 JAKE(I,J)=PWR(1,K)+(Y(1,K,K)*VEE(K)*VEE(K))
      JAKE(J,I)=PWR(1,K)-(Y(1,K,K)*VEE(K)*VEE(K))
   90 CONTINUE
C
C     BUILD AUGMENT VECTOR
C
      DO 100 I=1,NH
      K=JK(I)
  100 JAKE(I,NJP)=DP(K)
      DO 105 I=NBUS,NJAC
      K=JK(I-NH)
  105 JAKE(I,NJP)=DQ(K)
      CALL SOLVE (NJAC,JAKE)
      GO TO 40
  110 WRITE (6,203) ITER
      DO 47 I=1,NBUS
      ARC=ANG(I)*180.0/3.14159
      WRITE (10,204) I,ITYPE(I),VEE(I),ARC,P(I),DP(I),DQ(I)
   47 WRITE (6,204) I,ITYPE(I),VEE(I),ARC,P(I),DP(I),DQ(I)
      STOP
  200 FORMAT (2I5,3F10.5)
  201 FORMAT (2I4,5F8.5)
  202 FORMAT (2I5,10X,E10.4,I5)
```

Source Listing - Load Flow

```
203 FORMAT (/1X,'ITERATION NUMBER',I4/4X,'BUS',5X,'BUS',5X,'BUS',5X
   1,'BUS',10X,'POWER',11X,'MISMATCH'/2X,'NUMBER',3X,'TYPE',
   23X,'VOLTAGE',2X,'ANGLE',4X,2('REAL',3X,'REACTIVE',3X))
204 FORMAT (1X,I5,I8,3X,F8.4,F8.3,4F9.5)
205 FORMAT (1X,'FOR ITERATION NUMBER ',I3,' BUS',I4,' CHANGED TO ',
   1 'LOAD BUS. VAR DELIVERY SET AT ',F9.5)
    END


    SUBROUTINE SOLVE (N,A)
    DIMENSION A(240,241)
    MAX=N+1
    DO 12 K=1,N
    KP=MAX+1-K
    DO 10 J=1,KP
    JP=MAX+1-J
 10 A(K,JP)=A(K,JP)/A(K,K)
    DO 12 I=1,N
    IF (I.EQ.K) GO TO 12
    DO 12 J=1,KP
    JP=MAX+1-J
    A(I,JP)=A(I,JP)-A(I,K)*A(K,JP)
 12 CONTINUE
    RETURN
    END
```

Source Listing - Load Flow

179	118			1.0E-03	20					
1	1	0.955	0.0	0.510	0.270	0.0	0.0	-0.050	0.150	0.0
2	0	1.000	0.0	0.200	0.050	0.0	0.0	0.0	0.0	0.0
3	0	1.000	0.0	0.390	0.100	0.0	0.0	0.0	0.0	0.0
4	1	0.998	0.0	0.300	0.120	-0.090	0.0	-3.000	3.000	0.0
5	0	1.015	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-0.400
6	1	0.990	0.0	0.520	0.220	0.0	0.0	-0.130	0.500	0.0
7	0	1.000	0.0	0.190	0.020	0.0	0.0	0.0	0.0	0.0
8	1	1.015	0.0	0.0	0.0	-0.250	0.0	-2.000	3.000	0.0
9	0	1.000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	1	1.025	0.0	0.0	0.0	4.500	0.0	-1.470	2.000	0.0
11	0	1.000	0.0	0.700	0.230	0.0	0.0	0.0	0.0	0.0
12	1	1.000	0.0	0.470	0.100	0.850	0.0	-0.350	1.200	0.0
13	0	1.000	0.0	0.340	0.160	0.0	0.0	0.0	0.0	0.0
14	0	1.000	0.0	0.140	0.010	0.0	0.0	0.0	0.0	0.0
15	1	0.970	0.0	0.900	0.300	0.0	0.0	-0.100	0.300	0.0
16	0	1.000	0.0	0.250	0.100	0.0	0.0	0.0	0.0	0.0
17	0	1.000	0.0	0.110	0.030	0.0	0.0	0.0	0.0	0.0
18	1	0.973	0.0	0.600	0.340	0.0	0.0	-0.160	0.500	0.0
19	1	0.962	0.0	0.450	0.250	0.0	0.0	-0.080	0.240	0.0
20	0	1.000	0.0	0.180	0.030	0.0	0.0	0.0	0.0	0.0
21	0	1.000	0.0	0.140	0.080	0.0	0.0	0.0	0.0	0.0
22	0	1.000	0.0	0.100	0.050	0.0	0.0	0.0	0.0	0.0
23	0	1.000	0.0	0.070	0.030	0.0	0.0	0.0	0.0	0.0
24	1	0.992	0.0	0.0	0.0	-0.130	0.0	-3.000	3.000	0.0
25	1	1.020	0.0	0.0	0.0	2.200	0.0	-0.470	1.400	0.0
26	1	1.035	0.0	0.0	0.0	3.140	0.0	-10.000	10.000	0.0
27	1	0.968	0.0	0.620	0.130	-0.090	0.0	-3.000	3.000	0.0
28	0	1.000	0.0	0.170	0.070	0.0	0.0	0.0	0.0	0.0
29	0	1.000	0.0	0.240	0.040	0.0	0.0	0.0	0.0	0.0
30	0	1.000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
31	1	0.967	0.0	0.430	0.270	0.070	0.0	-3.000	3.000	0.0
32	1	0.963	0.0	0.590	0.230	0.0	0.0	-0.140	0.420	0.0
33	0	1.000	0.0	0.230	0.090	0.0	0.0	0.0	0.0	0.0
34	1	0.984	0.0	0.590	0.260	0.0	0.0	-0.080	0.240	0.140
35	0	1.000	0.0	0.330	0.090	0.0	0.0	0.0	0.0	0.0
36	1	0.980	0.0	0.310	0.170	0.0	0.0	-0.030	0.240	0.0
37	0	1.000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-0.250
38	0	1.000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
39	0	1.000	0.0	0.270	0.110	0.0	0.0	0.0	0.0	0.0
40	1	0.970	0.0	0.200	0.230	-0.460	0.0	-3.000	3.000	0.0
41	0	1.000	0.0	0.370	0.100	0.0	0.0	0.0	0.0	0.0
42	1	0.985	0.0	0.370	0.230	-0.590	0.0	-3.000	3.000	0.0
43	0	1.000	0.0	0.180	0.070	0.0	0.0	0.0	0.0	0.0
44	0	1.000	0.0	0.160	0.080	0.0	0.0	0.0	0.0	0.100
45	0	1.000	0.0	0.530	0.220	0.0	0.0	0.0	0.0	0.100
46	1	1.035	0.0	0.290	0.100	0.190	0.0	-1.000	1.000	0.100
47	0	1.000	0.0	0.340	0.0	0.0	0.0	0.0	0.0	0.0
48	0	1.000	0.0	0.200	0.110	0.0	0.0	0.0	0.0	0.150
49	1	1.035	0.0	0.870	0.300	2.040	0.0	-0.850	2.100	0.0
50	0	1.000	0.0	0.170	0.040	0.0	0.0	0.0	0.0	0.0
51	0	1.000	0.0	0.170	0.080	0.0	0.0	0.0	0.0	0.0
52	0	1.000	0.0	0.180	0.050	0.0	0.0	0.0	0.0	0.0
53	0	1.000	0.0	0.230	0.110	0.0	0.0	0.0	0.0	0.0
54	1	1.000	0.0	1.130	0.320	0.480	0.0	-3.000	3.000	0.0
55	1	1.000	0.0	0.630	0.220	0.0	0.0	-0.090	0.230	0.0
56	1	1.000	0.0	0.840	0.180	0.0	0.0	-0.080	0.150	0.0
57	0	1.000	0.0	0.120	0.030	0.0	0.0	0.0	0.0	0.0
58	0	1.000	0.0	0.120	0.030	0.0	0.0	0.0	0.0	0.0
59	1	1.035	0.0	2.770	1.130	1.550	0.0	-0.600	1.800	0.0
60	0	1.000	0.0	0.780	0.030	0.0	0.0	0.0	0.0	0.0
61	1	1.035	0.0	0.0	0.0	1.600	0.0	-1.000	3.000	0.0

Input Data

62	1	1.020	0.0	0.770	0.140	0.0	0.0	-0.200	0.200	0.0
63	0	1.000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
64	0	1.000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
65	1	1.035	0.0	0.0	0.0	3.910	0.0	-0.670	2.000	0.0
66	1	1.035	0.0	0.390	0.180	3.920	0.0	-0.670	2.000	0.0
67	0	1.000	0.0	0.280	0.070	0.0	0.0	0.0	0.0	0.0
68	0	1.000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
69	1	1.035	0.0	0.0	0.0	5.1	0.0	-0.670	2.000	0.0
70	1	0.984	0.0	0.660	0.200	0.0	0.0	-0.100	0.320	0.0
71	0	1.000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
72	1	0.980	0.0	0.0	0.0	-0.120	0.0	-1.000	1.000	0.0
73	1	0.991	0.0	0.0	0.0	-0.060	0.0	-1.000	1.000	0.0
74	1	0.980	0.0	0.680	0.270	0.0	0.0	-0.060	0.090	0.120
75	0	1.000	0.0	0.470	0.110	0.0	0.0	0.0	0.0	0.0
76	1	0.950	0.0	0.680	0.360	0.0	0.0	-0.090	0.230	0.0
77	1	1.006	0.0	0.610	0.280	0.0	0.0	-0.200	0.700	0.0
78	0	1.000	0.0	0.710	0.260	0.0	0.0	0.0	0.0	0.0
79	0	1.000	0.0	0.390	0.320	0.0	0.0	0.0	0.0	0.200
80	2	1.025	0.0	1.300	0.260	4.770	0.0	-1.650	2.800	0.0
81	0	1.000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
82	0	1.000	0.0	0.540	0.270	0.0	0.0	0.0	0.0	0.200
83	0	1.000	0.0	0.200	0.100	0.0	0.0	0.0	0.0	0.100
84	0	1.000	0.0	0.110	0.070	0.0	0.0	0.0	0.0	0.0
85	1	1.015	0.0	0.240	0.150	0.0	0.0	-0.080	0.230	0.0
86	0	1.000	0.0	0.210	0.100	0.0	0.0	0.0	0.0	0.0
87	1	1.015	0.0	0.0	0.0	0.0	0.0	-1.000	10.000	0.0
88	0	1.000	0.0	0.480	0.100	0.0	0.0	0.0	0.0	0.0
89	1	1.035	0.0	0.0	0.0	6.070	0.0	-2.100	3.000	0.0
90	1	0.985	0.0	0.780	0.420	-0.850	0.0	-3.000	3.000	0.0
91	1	0.980	0.0	0.0	0.0	-0.100	0.0	-1.000	1.000	0.0
92	1	1.020	0.0	0.650	0.100	0.0	0.0	-0.030	0.090	0.0
93	0	1.000	0.0	0.120	0.070	0.0	0.0	0.0	0.0	0.0
94	0	1.000	0.0	0.300	0.160	0.0	0.0	0.0	0.0	0.0
95	0	1.000	0.0	0.420	0.310	0.0	0.0	0.0	0.0	0.0
96	0	1.000	0.0	0.380	0.150	0.0	0.0	0.0	0.0	0.0
97	0	1.000	0.0	0.150	0.090	0.0	0.0	0.0	0.0	0.0
98	0	1.000	0.0	0.340	0.080	0.0	0.0	0.0	0.0	0.0
99	1	1.010	0.0	0.0	0.0	-0.420	0.0	-1.000	1.000	0.0
100	1	1.030	0.0	0.370	0.180	2.520	0.0	-0.500	1.550	0.0
101	0	1.000	0.0	0.220	0.150	0.0	0.0	0.0	0.0	0.0
102	0	1.000	0.0	0.050	0.030	0.0	0.0	0.0	0.0	0.0
103	1	1.015	0.0	0.230	0.160	0.400	0.0	-0.150	0.400	0.0
104	1	0.980	0.0	0.380	0.250	0.0	0.0	-0.050	0.230	0.0
105	1	0.980	0.0	0.310	0.260	0.0	0.0	-0.080	0.230	0.200
106	0	1.000	0.0	0.430	0.160	0.0	0.0	0.0	0.0	0.0
107	1	0.952	0.0	0.280	0.120	-0.220	0.0	-2.000	2.000	0.060
108	0	1.000	0.0	0.020	0.010	0.0	0.0	0.0	0.0	0.0
109	0	1.000	0.0	0.080	0.030	0.0	0.0	0.0	0.0	0.0
110	1	0.990	0.0	0.390	0.300	0.0	0.0	-0.080	0.230	0.060
111	1	1.035	0.0	0.0	0.0	0.360	0.0	-1.000	10.000	0.0
112	1	0.975	0.0	0.250	0.130	-0.430	0.0	-1.000	10.000	0.0
113	1	0.993	0.0	0.0	0.0	-0.060	0.0	-1.000	2.000	0.0
114	0	1.000	0.0	0.080	0.030	0.0	0.0	0.0	0.0	0.0
115	0	1.000	0.0	0.220	0.070	0.0	0.0	0.0	0.0	0.0
116	1	1.005	0.0	0.0	0.0	-1.240	0.0	-10.000	10.000	0.0
117	0	1.000	0.0	0.200	0.080	0.0	0.0	0.0	0.0	0.0
118	0	1.000	0.0	0.330	0.150	0.0	0.0	0.0	0.0	0.0

1	2	2.78030	-9.16674	0.02540
1	3	6.56766	-21.58673	0.01080
2	12	4.51227	-14.86397	0.01560
3	5	1.96818	-8.82007	0.02940
3	12	1.73212	-5.72603	0.04060
4	11	4.04236	-13.30691	0.01740
4	5	26.76974	-118.97678	0.00200
5	6	3.89193	-17.66084	0.01420
5	8	0.0	-37.45317	0.0
5	[illegible]	[illegible]	[illegible]	[illegible]

Input Data

[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
6	7	9.93619	-45.92726	0.00540
7	12	6.99209	-27.64316	0.00860
8	30	1.68057	-19.69789	0.51400
8	9	2.56408	-22.58513	1.16200
9	10	2.49137	-30.85474	1.23000
11	12	14.08216	-46.78142	0.00500
11	13	3.80369	-12.52475	0.01880
12	16	2.86293	-11.26267	0.02140
12	117	1.59072	-6.76904	0.02580
12	14	3.93720	-12.94697	0.01820
13	15	1.13994	-3.74463	0.06260
14	15	1.43148	-4.69142	0.05020
15	17	6.33419	-20.97000	0.04440
15	19	7.07397	-23.22621	0.01000
15	33	2.24595	-7.35252	0.03200
16	17	1.31605	-5.22072	0.04660
17	113	9.20289	-30.44032	0.00760
17	18	4.55296	-18.69302	0.01280
30	17	0.0	-25.77319	0.0
17	31	1.77685	-5.85910	0.04000
18	19	4.34663	-19.20533	0.01140
19	20	1.75928	-8.16809	0.02980
19	34	1.12804	-3.70515	0.06320
20	21	2.42612	-11.25562	0.02160
21	22	2.12273	-9.85191	0.02460
22	23	1.29297	-6.01120	0.04040
23	32	2.21694	-8.06351	0.11720
23	24	5.18654	-18.90205	0.04980
23	25	2.34821	-12.04210	0.08640
24	70	0.56848	-2.28895	0.10200
24	72	1.19615	-4.80422	0.04880
26	25	0.0	-26.17801	0.0
25	27	1.15300	-5.91003	0.17640
26	30	1.05921	-11.53061	0.90800
27	32	3.67892	-12.12918	0.01920
27	115	2.84733	-12.86509	0.01980
27	28	2.48858	-11.13998	0.02160
28	29	2.50683	-9.97443	0.02380
29	31	8.90905	-27.30460	0.00820
30	38	1.56614	-18.38509	0.42200
31	32	2.81390	-9.30097	0.02500
32	113	1.36693	-4.51199	0.05180
32	114	3.43713	-15.52168	0.01620
33	37	1.89617	-6.48809	0.03660
34	36	10.95815	-33.75613	0.00560
34	37	27.33389	-98.82259	0.00980
34	43	1.37835	-5.61020	0.04220
35	36	20.20570	-93.68112	0.00260
35	37	4.24532	-19.18111	0.01320
38	37	0.0	-26.66666	0.0
37	39	2.61690	-8.64149	0.02700
37	40	1.86828	-5.29292	0.04200
38	65	0.91809	-10.05819	1.04600
39	40	4.60137	-15.12950	0.01540
40	41	5.61593	-18.86177	0.01220
40	42	1.51767	-5.00421	0.04660
41	42	2.05968	-6.78188	0.03440
42	49	1.31605	-5.91855	0.17200
43	44	0.95122	-3.83931	0.06060
44	45	2.59868	-10.45272	0.02240
45	46	2.00126	-6.78429	0.03320
45	49	1.74158	-4.73589	0.04440
46	47	2.16241	-7.22700	0.03160
46	48	1.52798	-4.80512	0.04720
47	49	4.47196	-14.63338	0.01600
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

Input Data

[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
48	49	6.23550	-17.59177	0.01260
49	50	4.19288	-11.80917	0.01860
49	51	2.29994	-6.48338	0.03420
49	66	4.11370	-20.97987	0.04960
49	69	0.85892	-2.82530	0.08280
49	54	1.76036	-6.41337	0.14680
50	57	2.34621	-6.63276	0.03320
51	52	5.24611	-15.19563	0.01400
51	58	4.38155	-12.35425	0.01780
52	53	1.42744	-5.76263	0.04040
53	54	1.68853	-7.83272	0.03100
54	55	3.19828	-13.37977	0.02020
54	56	27.68094	-97.39595	0.00720
54	59	0.91274	-4.16058	0.05980
55	56	19.11969	-60.14740	0.00380
55	59	0.96512	-4.42150	0.05640
56	57	3.26416	-9.19296	0.02420
56	58	3.26416	-9.19296	0.02420
56	59	2.53480	-7.47362	0.11040
59	60	1.43895	-6.58197	0.03760
59	61	1.39125	-6.36245	0.03880
60	61	13.75589	-71.42482	0.01460
60	62	3.72896	-17.00772	0.01460
61	62	5.53680	-25.38824	0.00980
64	61	0.0	-37.31343	0.0
62	66	0.96695	-4.37336	0.05780
62	67	1.79733	-8.15068	0.03100
63	64	4.21952	-49.64137	0.21600
63	59	0.0	-25.90672	0.0
64	65	2.93692	-32.85001	0.38000
65	68	5.42720	-62.02512	0.63800
65	66	0.0	-27.02702	0.0
66	67	2.07331	-9.39467	0.02680
68	69	0.0	-27.02702	0.0
68	81	4.13696	-45.15681	0.80800
68	116	18.64508	-248.60162	0.16400
69	75	2.45094	-7.38309	0.12400
69	77	2.76986	-9.05358	0.10380
69	70	1.76170	-7.45787	0.12200
70	71	6.57850	-26.53827	0.00860
70	74	2.09823	-6.92261	0.03360
70	75	1.97118	-6.49395	0.03600
71	72	1.29692	-5.23421	0.04440
71	73	4.07141	-21.24623	0.01180
74	75	6.83466	-22.55954	0.01020
75	118	5.74517	-19.05811	0.01180
75	77	1.37933	-4.58781	0.04980
76	118	5.08004	-16.85086	0.01360
76	77	20.27026	-6.75676	0.03680
77	78	22.09615	-74.05199	0.01260
77	80	8.90905	-27.30460	0.07000
77	82	3.65011	-10.44814	0.08180
78	79	8.64664	-39.07002	0.00640
79	80	3.00028	-13.53972	0.01860
80	96	1.03514	-5.29203	0.04940
80	97	2.02021	-10.31082	0.02540
80	98	1.94596	-8.83043	0.02860
80	99	1.02029	-4.62951	0.05460
81	80	0.0	-27.02702	0.0
82	96	5.27440	-17.25575	0.05440
82	83	7.64505	-24.98294	0.03800
83	84	2.93011	-6.18840	0.02580
83	85	1.81030	-6.23079	0.03480
84	85	6.01491	-12.76676	0.01220
85	86	2.14015	-7.52113	0.02760
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

Input Data

[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
85	89	0.78360	-5.67210	0.04700
86	87	0.64369	-4.73408	0.04440
88	89	2.64125	-13.52930	0.01920
89	90	3.50043	-14.46698	0.15880
89	92	5.24426	-25.22554	0.09620
90	91	3.32717	-10.95084	0.02140
91	92	2.18922	-7.19558	0.03260
92	93	3.28383	-10.79337	0.02180
92	94	1.76335	-5.79230	0.04060
92	100	0.71034	-3.23380	0.07720
92	102	3.75446	-17.06297	0.01460
93	94	3.80837	-12.50101	0.01380
94	95	6.41462	-21.09048	0.01100
94	96	3.25067	-10.50124	0.02300
94	100	4.83585	-15.75728	0.06040
95	96	5.20627	-16.65398	0.01480
96	97	2.12752	-10.88355	0.02400
98	100	1.18095	-5.32467	0.04760
99	100	2.59602	-11.72536	0.02160
100	101	1.65930	-7.55972	0.03280
100	103	5.31164	-17.42882	0.05360
100	104	1.03322	-4.67354	0.05400
100	106	1.07841	-4.08191	0.06200
101	102	1.67084	-8.51766	0.02940
103	110	1.13668	-5.27058	0.04600
103	104	1.70933	-5.81026	0.04060
103	105	1.82790	-5.55204	0.04080
104	105	6.48393	-24.75684	0.00980
105	106	4.39134	-17.15759	0.01440
105	107	1.46014	-5.04160	0.04720
105	108	4.64140	-12.50156	0.01840
106	107	1.46014	-5.04160	0.04720
108	109	11.17390	-30.64841	0.00760
109	110	4.22539	-11.58182	0.02020
110	111	3.55742	-12.20843	0.02000
110	112	5.24852	-13.59941	0.06200
114	115	20.27324	-91.67038	0.00280

Input Data

```
FOR ITERATION NUMBER  12 BUS   1 CHANGED TO LOAD BUS. VAR DELIVERY SET AT  -0.05
FOR ITERATION NUMBER  12 BUS  12 CHANGED TO LOAD BUS. VAR DELIVERY SET AT   1.20.
FOR ITERATION NUMBER  12 BUS  19 CHANGED TO LOAD BUS. VAR DELIVERY SET AT  -0.08
FOR ITERATION NUMBER  12 BUS  62 CHANGED TO LOAD BUS. VAR DELIVERY SET AT  -0.29.
FOR ITERATION NUMBER  12 BUS  70 CHANGED TO LOAD BUS. VAR DELIVERY SET AT  -0.19
FOR ITERATION NUMBER  12 BUS  74 CHANGED TO LOAD BUS. VAR DELIVERY SET AT   0.09
FOR ITERATION NUMBER  12 BUS  76 CHANGED TO LOAD BUS. VAR DELIVERY SET AT   0.23
FOR ITERATION NUMBER  12 BUS  77 CHANGED TO LOAD BUS. VAR DELIVERY SET AT  -0.20.
FOR ITERATION NUMBER  12 BUS  92 CHANGED TO LOAD BUS. VAR DELIVERY SET AT   0.09
FOR ITERATION NUMBER  12 BUS 103 CHANGED TO LOAD BUS. VAR DELIVERY SET AT   0.40
FOR ITERATION NUMBER  12 BUS 104 CHANGED TO LOAD BUS. VAR DELIVERY SET AT  -0.08
FOR ITERATION NUMBER  12 BUS 110 CHANGED TO LOAD BUS. VAR DELIVERY SET AT  -0.09
```

ITERATION NUMBER 16

BUS NUMBER	BUS TYPE	BUS VOLTAGE	BUS ANGLE	POWER REAL	POWER REACTIVE	MISMATCH REAL	MISMATCH REACTIVE
1	0	0.9557	-16.959	-0.51301	-0.32000	0.00001	0.00000
2	0	0.9745	-16.455	-0.20000	-0.09000	-0.00000	-0.00000
3	0	0.9682	-16.065	-0.39000	-0.10002	-0.00000	0.00002
4	1	0.9980	-12.331	-0.38996	0.02955	-0.00004	0.0
5	0	0.9994	-11.848	-0.00004	0.00009	0.00004	-0.00009
6	1	0.9900	-14.633	-0.52000	-0.09199	-0.00000	0.0
7	0	0.9910	-15.105	-0.19001	-0.01997	0.00001	-0.00003
8	1	1.0150	-6.715	-0.28000	-0.18718	0.0	0.0
9	0	1.0351	0.628	-0.00000	-0.00000	0.00000	0.00000
10	1	1.0350	8.410	4.49999	-0.71799	0.00001	0.0
11	0	0.9874	-14.957	-0.70000	-0.22998	0.00000	-0.00002
12	0	0.9945	-15.512	0.38001	1.10000	-0.00001	-0.00000
13	0	0.9701	-16.327	-0.34000	-0.15999	-0.00000	-0.00001
14	0	0.9869	-16.225	-0.14000	-0.00999	-0.00000	-0.00001
15	1	0.9700	-16.523	-0.89998	-0.20394	-0.00002	0.0
16	0	0.9862	-15.796	-0.25000	-0.10000	0.00000	0.00000
17	0	0.9925	-13.953	-0.11001	-0.02992	0.00001	-0.00008
18	1	0.9730	-16.224	-0.60000	-0.02142	-0.00000	0.0
19	0	0.9629	-16.715	-0.44999	-0.32994	-0.00001	-0.00006
20	0	0.9554	-15.895	-0.18000	-0.02999	-0.00000	-0.00001
21	0	0.9543	-14.341	-0.14000	-0.08000	-0.00000	-0.00000
22	0	0.9626	-11.811	-0.10000	-0.04999	-0.00000	-0.00001
23	0	0.9907	-6.898	-0.07000	-0.02996	0.00000	-0.00004
24	1	0.9920	-7.110	-0.13000	0.34657	-0.00000	0.0
25	1	1.0200	0.331	2.20000	-0.15945	0.00000	0.0
26	1	1.0350	2.047	3.14000	0.18752	0.0	0.0
27	1	0.9680	-12.711	-0.71000	0.09562	0.0	0.0
28	0	0.9616	-14.387	-0.17000	-0.07000	0.00000	0.00000
29	0	0.9632	-15.326	-0.24001	-0.03997	0.00001	-0.00003
30	0	1.0141	-8.763	0.00000	0.00006	-0.00000	-0.00006
31	1	0.9670	-15.190	-0.35999	0.07237	-0.00001	0.0
32	1	0.9630	-13.200	-0.59999	-0.31798	-0.00001	0.0
33	0	0.9695	-17.123	-0.23000	-0.08[illegible]	0.00000	-0.00001
34	1	0.9840	-16.487	-0.59000	-0.19475	0.00000	0.0
35	0	0.9799	-16.933	-0.32997	-0.08995	-0.00003	-0.00005
36	1	0.9800	-16.940	-0.31002	-0.04452	0.00002	0.0
37	0	0.9877	-15.983	0.00001	0.00010	-0.00001	-0.00010
38	0	1.0098	-10.719	-0.00000	-0.00006	0.00000	0.00006
39	0	0.9688	-19.443	-0.27000	-0.11001	0.00000	0.00001
40	1	0.9700	-20.558	-0.66001	0.09043	0.00001	0.0
41	0	0.9668	-21.011	-0.37000	-0.10002	0.00000	0.00002
42	1	0.9850	-19.434	-0.96000	0.11363	0.00000	0.0
43	0	0.9841	-16.643	-0.18000	-0.07000	0.00000	-0.00000
44	0	1.0013	-14.359	-0.16000	-0.08000	-0.00000	-0.00000
45	0	1.0063	-12.654	-0.53000	-0.22000	-0.00000	0.00000

Output

46	1	1.0350	-10.132	-0.05000	0.12590	-0.00000	0.0
47	0	1.0315	-7.723	-0.34000	-0.00002	0.00000	0.00002
48	0	1.0350	-8.481	-0.20000	-0.11001	0.00000	0.00001
49	1	1.0350	-7.416	1.17001	0.78816	-0.00001	0.0
50	0	1.0202	-9.447	-0.17001	-0.34002	0.00001	0.00002
51	0	0.9974	-12.001	-0.17000	-0.07999	0.00000	-0.00001
52	0	0.9903	-12.916	-0.18000	-0.05000	-0.00000	-0.00000
53	0	0.9867	-13.840	-0.23000	-0.11000	0.00000	0.00000
54	1	1.0000	-13.308	-0.65001	-0.20261	0.00001	0.0
55	1	1.0000	-13.296	-0.63000	-0.03015	0.00000	0.0
56	1	1.0000	-13.107	-0.84000	-0.02815	-0.00000	0.0
57	0	1.0055	-11.936	-0.12000	-0.02999	-0.00000	-0.00001
58	0	0.9963	-12.750	-0.12000	-0.03002	0.00000	0.00002
59	1	1.0350	-9.127	-1.22000	0.46130	-0.00000	0.0
60	0	1.0315	-5.481	-0.77999	-0.03004	-0.00001	0.00004
61	1	1.0350	-4.660	1.60000	0.10046	0.00000	0.0
62	0	1.0243	-5.124	-0.77000	-0.33999	0.00000	-0.00001
63	0	1.0349	-5.849	-0.00000	-0.00006	0.00000	0.00006
64	0	1.0366	-4.147	-0.00000	0.00002	0.00000	-0.00002
65	1	1.0350	-0.960	3.91000	0.10471	-0.00000	0.0
66	1	1.0350	-0.843	3.53000	-0.42006	0.00000	0.0
67	0	1.0238	-3.622	-0.28000	-0.07000	-0.00000	0.00000
68	0	1.0172	-1.040	0.00000	-0.00008	-0.00000	0.00008
69	1	1.0350	1.582	5.10000	0.64414	-0.00000	0.0
70	0	0.9826	-5.209	-0.66001	-0.30004	0.00001	0.00004
71	0	0.9862	-5.663	0.00000	0.00003	-0.00000	-0.00003
72	1	0.9800	-6.937	-0.12000	-0.10716	0.00000	0.0
73	1	0.9910	-5.883	-0.06000	0.11170	-0.00000	0.0
74	0	0.9682	-5.977	-0.68000	-0.18001	0.00000	0.00001
75	0	0.9747	-4.573	-0.47000	-0.10999	-0.00000	-0.00001
76	0	0.9579	-3.338	-0.68000	-0.13001	0.00000	0.00001
77	0	1.0027	-2.424	-0.61000	-0.48003	-0.00000	0.00003
78	0	0.9999	-2.710	-0.71000	-0.26000	-0.00000	-0.00000
79	0	1.0053	-2.370	-0.39001	-0.32002	0.00001	0.00002
80	2	1.0350	0.0	0.0	0.0	0.0	0.0
81	0	1.0293	-0.704	-0.00000	-0.00005	0.00000	0.00005
82	0	0.9988	-2.083	-0.54000	-0.27000	-0.00000	-0.00000
83	0	1.0010	-1.046	-0.20000	-0.10001	0.00000	0.00001
84	0	1.0055	1.173	-0.11000	-0.36999	0.00000	-0.00001
85	1	1.0150	2.540	-0.23999	0.02569	-0.00001	0.0
86	0	1.0053	1.121	-0.21000	-0.10000	0.00000	0.00000
87	1	1.0150	1.047	0.00000	0.02446	-0.00000	0.0
88	0	1.0178	5.578	-0.48000	-0.10002	0.00000	0.00002
89	1	1.0350	9.461	6.07001	0.55393	-0.00001	0.0
90	1	0.9850	3.726	-1.63000	-0.31908	-0.00000	0.0
91	1	0.9800	3.825	-0.10000	-0.32553	0.00000	0.0
92	0	1.0152	4.035	-0.64999	-0.01000	-0.00001	0.00000
93	0	1.0052	1.259	-0.12000	-0.07001	0.00000	0.00001
94	0	1.0045	-0.726	-0.29999	-0.16000	-0.00001	0.0
95	0	0.9926	-1.624	-0.42000	-0.30998	-0.00000	-0.00002
96	0	1.0013	-1.725	-0.38000	-0.15001	0.00000	0.00001
97	0	1.0133	-1.218	-0.15000	-0.08999	0.00000	-0.00001
98	0	1.0253	-1.693	-0.34000	-0.08001	0.00000	0.00001
99	1	1.0100	-2.082	-0.42000	-0.31259	0.00000	0.0
100	1	1.0300	-1.297	2.15001	1.06086	-0.00001	0.0
101	0	1.0102	0.131	-0.22000	-0.15001	0.00000	0.00001
102	0	1.0124	2.639	-0.05000	-0.02997	-0.00000	-0.00003
103	0	1.0144	-4.817	0.16999	0.23999	0.00001	0.00001
104	0	0.9812	-7.431	-0.38000	-0.32999	-0.00000	-0.00001
105	1	0.9800	-8.577	-0.31001	-0.15994	0.00001	0.0
106	0	0.9732	-8.778	-0.43000	-0.15999	-0.00000	-0.00001
107	1	0.9520	-11.335	-0.50000	-0.19716	-0.00000	0.0
108	0	0.9820	-9.770	-0.02000	-0.01000	-0.00000	-0.00000
109	0	0.9845	-10.219	-0.08000	-0.03001	0.00000	0.00001
110	0	0.9923	-11.065	-0.39000	-0.38001	-0.00000	0.00001
111	1	1.0350	-10.140	0.36000	0.47155	-0.00000	0.0
112	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

Output

112	1	0.9750	-13.[illegible]	-0.0[illegible]000	-0.[illegible]	0.00000	0.0
113	1	0.9930	-14.035	-0.06000	0.15381	0.00000	0.0
114	0	0.9601	-13.557	-0.07999	-0.03002	-0.00001	0.00002
115	0	0.9600	-13.569	-0.22000	-0.06994	-0.00000	-0.00006
116	1	1.0050	-1.403	-1.83998	-3.01552	-0.00002	0.0
117	0	0.9784	-17.039	-0.20000	-0.08000	0.00000	0.00000
118	0	0.9604	-4.453	-0.33000	-0.15002	0.00000	0.00002

Output

APPENDIX B

This appendix contains the FORTRAN source listing, a description of the input, and an output list for the system reduction segment of the program. The input is read from two separate disk files. The first is the same file used as input to the load flow segment and described in Appendix A. The second file may be recognized as the last part of the output list shown in Appendix A, with a second bus type code added. A typical line is:

```
 99   2   1     1.0100  -2.082. . .
(A)  (B) (C)      (D)      (E)
```

where

(A) is the bus number

(B) is the reduction code: 0 (or blank) for eliminate, 1 for boundary and 2 for retain

(C) is the same code as in the load flow.

(D) is the base case bus voltage magnitude

and (E) is the base case bus voltage angle

the remainder of the information on the line is not needed.

The output is an echo print of pertinent portions of input data, followed by the 9 x 9 Jacobian correction matrix. The last section is a list of conditions at the boundary buses, showing bus number, bus voltage magnitude and angle (in radians), and the real and reactive power injected.

The last item in this Appendix is a source listing of the load flow segment modified to handle the reduced system.

```
C
C     NERD118.FORT. NOV. 29. '79
C     BUILD JACOBIAN. REORDER. PERFORM TRUNCATED BIFACTORIZATION
C     READS NADM. NBUS. ADMITTANCES AND Y-SHUNT ON TUBE 1.
C     READS BUS TYPES AND BASE CASE VOLTAGES AND ANGLES ON TUBE 9.
C     WRITES THE REDUCED JACOBIAN 'JCOR'. POWER INJECTIONS. AND
C     BASE CASE VOLTAGES AND ANGLES ON TUBE 10.
C
      COMPLEX WYE(120.120).V(120).CUR(120).P(120).PINJ(10)
      COMPLEX CONJG.WHY.CINJ
      DIMENSION Y(2.120.120).E(2.120).AYE(2.120).PWR(2.120)
      DIMENSION ITYPE(120).JTYPE(120).VEE(120).ANG(120).YSH(120)
      DIMENSION KINJ(10).JK(240).NK(240)
      EQUIVALENCE (WYE(1.1).Y(1.1.1)).(V(1).E(1.1)).(CUR(1).AYE(1.1))
      EQUIVALENCE (P(1).PWR(1.1))
      REAL JAKE(240.240)
C
C     READ PARAMETERS AND INITIALIZE
C
      READ (1.202)NADM.NBUS
      DO 10 I=1.NBUS
      DO 10 J=1.NBUS
   10 WYE(I.J)=CMPLX(0.0.0.0)
C
C     READ BUS DATA
C
      DO 35 I=1.NBUS
      READ (9.201) IBUS.JTYPE(I).ITYPE(I).VEE(I).ANG(I)
      ANG(I)=ANG(I)*3.14159/180.0
      E(1.I)=VEE(I)*COS(ANG(I))
      E(2.I)=VEE(I)*SIN(ANG(I))
      READ (1.204) YSH(I)
      WRITE (6.205) IBUS.JTYPE(I).ITYPE(I).VEE(I).ANG(I).YSH(I)
   35 IF(JTYPE(I).EQ.0) Y(2.I.I)=Y(2.I.I)+YSH(I)
C
C     BUILD ADMITTANCE MATRIX
C
      DO 20 I=1.NADM
      READ (1.200) IFR.ITO.WHY.CHG
      WYE(IFR.ITO)=-WHY
      WYE(ITO.IFR)=-WHY
      Y(2.IFR.IFR)=Y(2.IFR.IFR)+(CHG/2.0)
   20 Y(2.ITO.ITO)=Y(2.ITO.ITO)+(CHG/2.0)
      DO 30 I=1.NBUS
      DO 30 J=1.NBUS
   30 IF(I.NE.J) WYE(I.I)=WYE(I.I)-WYE(I.J)
      K=0
      DO 36 I=1.NBUS
C
C     CALCULATE POWER INJECTIONS
C
      IF(JTYPE(I).NE.1.OR.ITYPE(I).EQ.2) GO TO 36
      PINJ(I)=CMPLX(0.0.0.0)
      CINJ=CMPLX(0.0.0.0)
      DO 38 J=1.NBUS
      IF(JTYPE(J).NE.0) GO TO 38
      IF(I.NE.J) CINJ=CINJ+(V(I)-V(J))*WYE(I.J)
   38 CONTINUE
```

```
      K=K+1
      PINJ(K)=V(I)*CONJG(CINJ)
      KINJ(K)=I
   36 CONTINUE
      KK=K
C
C     CALCULATE CURRENTS AND POWERS
C
      DO 45 I=1,NBUS
   45 CUR(I)=CMPLX(0.0,0.0)
      DO 50 I=1,NBUS
      IF (ITYPE(I).EQ.2.OR.JTYPE(I).EQ.2) GO TO 50
      DO 52 J=1,NBUS
   52 CUR(I)=CUR(I)+WYE(I,J)*V(J)
      P(I)=V(I)*CONJG(CUR(I))
   50 CONTINUE
C
C     COUNT JACOBIAN
C
      NLD=0
      NVC=0
      DO 60 I=1,NBUS
      J=ITYPE(I)-1
      IF (J) 61,62,60
   61 NLD=NLD+1
      GO TO 60
   62 NVC=NVC+1
   60 CONTINUE
      NJAC=2*NLD+NVC
      NH=NLD+NVC
C
C     REORDER BUSSES
C
      K=0
      L=NLD
      DO 70 I=1,NBUS
      J=ITYPE(I)-1
      IF (J) 71,72,70
   71 K=K+1
      JK(K)=I
      GO TO 70
   72 L=L+1
      JK(L)=I
   70 CONTINUE
C
C     COUNT REDUCE
C
      NL1=0
      NV1=0
      NL2=0
      NV2=0
      DO 160 I=1,NBUS
      IF (JTYPE(I).EQ.1) ITYPE(I)=ITYPE(I)+3
      IF (JTYPE(I).EQ.2) ITYPE(I)=ITYPE(I)+6
      J=ITYPE(I)+1
      GO TO (161,162,160,163,164,160,160,160,160),J
  161 NL1=NL1+1
      GO TO 160
```

Source Listing - Reduction

```
  162 NV1=NV1+1
      GO TO 160
  163 NL2=NL2+1
      GO TO 160
  164 NV2=NV2+1
  160 CONTINUE
      NJ1=2*NL1+NV1
      NJ2=2*NL2+NV2
      NJAR=NJ1+NJ2
      NH1=NL1+NV1
      NH2=NL2+NV2
      NJ1P1=NJ1+1
C
C     REORDER FOR REDUCE
C
      K=0
      L=NL1
      M=NJ1
      N=M+NL2
      DO 170 I=1,NH
      JKI=JK(I)
      J=ITYPE(JKI)+1
      GO TO (171,172,175,173,174,175,176,175,175),J
  171 K=K+1
      NK(I)=K
      NK(I+NH)=K+NH1
      GO TO 170
  172 L=L+1
      NK(I)=L
      GO TO 170
  173 M=M+1
      NK(I)=M
      NK(I+NH)=M+NH2
      GO TO 170
  174 N=N+1
      NK(I)=N
      GO TO 170
  176 NK(I+NH)=NJAC
  175 NK(I)=NJAC
  170 CONTINUE
C
C     BUILD JACOBIAN
C
      DO 80 I=1,NH
      K=JK(I)
      NKI=NK(I)
      IF (I.LE.NLD) NKIP=NK(I+NH)
      DO 80 J=1,NH
      M=JK(J)
      NKJ=NK(J)
      IF (J.LE.NLD) NKJP=NK(J+NH)
      IF (K.EQ.M) GO TO 85
      A=E(1,M)*Y(1,K,M)-E(2,M)*Y(2,K,M)
      B=E(2,M)*Y(1,K,M)+E(1,M)*Y(2,K,M)
      JAKE(NKI,NKJ)=(A*E(2,K))-(B*E(1,K))
      IF (I.LE.NLD.AND.J.LE.NLD) JAKE(NKIP,NKJP)=JAKE(NKI,NKJ)
      GO TO 80
   85 JAKE(NKI,NKJ)=-PWR(2,K)-(Y(2,K,K)*VEE(K)*VEE(K))
```

Source Listing - Reduction

```
      IF(I.LE.NLD.AND.J.LE.NLD)JAKE(NKIP.NKJP)=JAKE(NKI.NKJ)+2.*PWR(2.K;
   80 CONTINUE
      DO 90 I=1.NH
      K=JK(I)
      NKI=NK(I)
      NKIP=NK(I+NH)
      DO 90 J=NBUS.NJAC
      NKJ=NK(J)
      NKJM=NK(J-NH)
      M=JK(J-NH)
      IF (K.EQ.M) GO TO 95
      A=E(1.M)*Y(1.K.M)-E(2.M)*Y(2.K.M)
      B=E(2.M)*Y(1.K.M)+E(1.M)*Y(2.K.M)
      JAKE(NKI.NKJ)=(A*E(1.K))+(B*E(2.K))
      JAKE(NKIP.NKJM)=-JAKE(NKI.NKJ)
      GO TO 90
   95 JAKE(NKI.NKJ)=PWR(1.K)+(Y(1.K.K)*VEE(K)*VEE(K))
      JAKE(NKJ.NKI)=PWR(1.K)-(Y(1.K.K)*VEE(K)*VEE(K))
   90 CONTINUE
C
C     PERFORM BIFACTORIZATION
C
      DO 29 IP=1.NJ1
      IPP=IP+1
      DO 39 I=IPP.NJAR
   39 JAKE(IP.I)=JAKE(IP.I)/JAKE(IP.IP)
      DO 29 I=IPP.NJAR
      DO 29 J=IPP.NJAR
   29 JAKE(I.J)=JAKE(I.J)-JAKE(I.IP)*JAKE(IP.J)
      WRITE (6.206)
      DO 121 I=IPP.NJAR
      WRITE (6.203) (JAKE(I.J).J=IPP.NJAR)
  121 WRITE (10.213) (JAKE(I.J).J=IPP.NJAR)
      WRITE (6.206)
      DO 122 I=1.KK
      KI=KINJ(I)
      WRITE (6.207) KINJ(I).PINJ(I).VEE(KI).ANG(KI)
  122 WRITE (10.217) KINJ(I).PINJ(I).VEE(KI).ANG(KI)
      STOP
  200 FORMAT (2I5.3F10.5)
  201 FORMAT (I6.2I4.3X.F8.4.F8.3)
  202 FORMAT (2I5)
  203 FORMAT (1X.12F10.5)
  213 FORMAT (12F10.5)
  204 FORMAT (72X.F8.5)
  205 FORMAT (3I6.3F12.4)
  206 FORMAT (72X)
  207 FORMAT (1X.I10.4F12.4)
  217 FORMAT (I10.4F12.4)
      END
```

Source Listing - Reduction

1		0.9557	-16.950	-0.51001	-0.32000	0.00001	0.00000
2		0.9745	-16.455	-0.20000	-0.09000	-0.00000	-0.00000
3		0.9682	-16.065	-0.39000	-0.10002	-0.00000	0.00002
4	1	0.9980	-12.331	-0.38996	0.03055	-0.00004	0.0
5		0.9994	-11.848	-0.00004	0.00009	0.00004	-0.00009
6	1	0.9900	-14.633	-0.52000	-0.09159	-0.00000	0.0
7		0.9910	-15.105	-0.15001	-0.01997	0.00001	-0.00003
8	1	1.0150	-6.715	-0.28000	-0.18718	0.0	0.0
9		1.0351	0.628	-0.00000	-0.00000	0.00000	0.00000
10	1	1.0350	8.410	4.49995	-0.71799	0.00001	0.0
11		0.9874	-14.957	-0.70000	-0.22998	0.00000	-0.00002
12		0.9945	-15.512	0.38001	1.10000	-0.00001	-0.00000
13		0.9701	-16.337	-0.34000	-0.15999	-0.00000	-0.00001
14		0.9869	-16.225	-0.14000	-0.00999	-0.00000	-0.00001
15	1	0.9700	-16.523	-0.89998	-0.20394	-0.00002	0.0
16		0.9862	-15.796	-0.25000	-0.10000	0.00000	0.00000
17		0.9925	-13.953	-0.11001	-0.02992	0.00001	-0.00008
18	1	0.9730	-16.224	-0.60000	-0.02142	-0.00000	0.0
19		0.9629	-16.715	-0.44995	-0.32994	-0.00001	-0.00006
20		0.9554	-15.895	-0.18000	-0.02999	-0.00000	-0.00001
21		0.9543	-14.341	-0.14000	-0.08000	-0.00000	-0.00000
22		0.9636	-11.811	-0.10000	-0.04999	-0.00000	-0.00001
23		0.9907	-6.898	-0.07000	-0.02996	0.00000	-0.00004
24	1	0.9920	-7.110	-0.13000	0.04657	-0.00000	0.0
25	1	1.0200	0.331	2.20000	-0.15945	0.00000	0.0
26	1	1.0350	2.047	3.14000	0.18752	0.0	0.0
27	1	0.9680	-12.711	-0.71000	0.09562	0.0	0.0
28		0.9616	-14.387	-0.17000	-0.07000	0.00000	0.00000
29		0.9632	-15.326	-0.24001	-0.03997	0.00001	-0.00003
30		1.0141	-8.763	0.00000	0.00006	-0.00000	-0.00006
31	1	0.9670	-15.190	-0.35995	0.07237	-0.00001	0.0
32	1	0.9630	-13.200	-0.58999	-0.31798	-0.00001	0.0
33		0.9695	-17.123	-0.23000	-0.08999	0.00000	-0.00001
34	1	0.9840	-16.487	-0.59000	-0.19975	0.00000	0.0
35		0.9799	-16.933	-0.32997	-0.08995	-0.00003	-0.00005
36	1	0.9800	-16.940	-0.31002	-0.04452	0.00002	0.0
37		0.9877	-15.983	0.00001	0.00010	-0.00001	-0.00010
38		1.0098	-10.719	-0.00000	-0.00006	0.00000	0.00006
39		0.9688	-19.443	-0.27000	-0.11001	0.00000	0.00001
40	1	0.9700	-20.558	-0.66001	0.09043	0.00001	0.0
41		0.9668	-21.011	-0.37000	-0.10002	0.00000	0.00002
42	1	0.9850	-19.434	-0.96000	0.11363	0.00000	0.0
43		0.9841	-16.643	-0.18000	-0.07000	0.00000	-0.00000
44		1.0013	-14.369	-0.16000	-0.08000	-0.00000	-0.00000
45		1.0063	-12.654	-0.53000	-0.22000	-0.00000	0.00000
46	1	1.0350	-10.133	-0.09000	0.12850	-0.00000	0.0
47		1.0315	-7.723	-0.34000	-0.00002	0.00000	0.00002
48		1.0350	-8.481	-0.20000	-0.11001	0.00000	0.00001
49	1	1.0350	-7.416	1.17001	0.78816	-0.00001	0.0
50		1.0202	-9.447	-0.17001	-0.04002	0.00001	0.00002
51		0.9974	-12.001	-0.17000	-0.07999	0.00000	-0.00001
52		0.9903	-12.916	-0.18000	-0.05000	-0.00000	-0.00003
53		0.9867	-13.840	-0.23000	-0.11000	0.00000	0.00000
54	1	1.0000	-13.008	-0.65001	-0.20261	0.00001	0.0
55	1	1.0000	-13.296	-0.63000	-0.03015	0.00000	0.0
56	1	1.0000	-13.107	-0.84000	-0.08815	-0.00000	0.0
57		1.0055	-11.936	-0.12000	-0.02999	-0.00000	-0.00001
58		0.9963	-12.750	-0.12000	-0.03002	0.00000	0.00002
59	1	1.0350	-9.127	-1.22000	0.46130	-0.00000	0.0
60		1.0315	-5.481	-0.77999	-0.03004	-0.00001	0.00004
61	1	1.0350	-4.660	1.60000	0.10046	0.00000	0.0
62		1.0243	-5.124	-0.77000	-0.33999	0.00000	-0.00001
63		1.0349	-5.849	-0.00000	-0.00006	0.00000	0.00006
[illegible]		[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

Input Data

[illegible]			[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
65		1	1.0350	-0.960	3.91000	0.10471	-0.00000	0.0
66		1	1.0350	-0.843	3.53000	-0.43006	0.00000	0.0
67			1.0238	-3.622	-0.28000	-0.07000	-0.00000	0.00000
68	1		1.0172	-1.040	0.00000	-0.00008	-0.00000	0.00008
69		1	1.0350	1.582	5.10000	0.64414	-0.00000	0.0
70			0.9826	-5.209	-0.66001	-0.30004	0.00001	0.00004
71			0.9862	-5.663	0.00000	0.00003	-0.00000	-0.00003
72		1	0.9800	-6.937	-0.12000	-0.10716	0.00000	0.0
73		1	0.9910	-5.883	-0.06000	0.11170	-0.00000	0.0
74			0.9682	-5.977	-0.68000	-0.18001	0.00000	0.00001
75			0.9747	-4.573	-0.47000	-0.10999	-0.00000	-0.00001
76			0.9579	-3.338	-0.68000	-0.13001	0.00000	0.00001
77	1		1.0027	-2.424	-0.61000	-0.48003	-0.00000	0.00003
78	2		0.9999	-2.710	-0.71000	-0.26000	-0.00000	-0.00000
79	2		1.0053	-2.370	-0.39001	-0.32002	0.00001	0.00002
80	2	2	1.0350	0.0	0.0	0.0	0.0	0.0
81	2		1.0293	-0.704	-0.00000	-0.00005	0.00000	0.00005
82	1		0.9988	-2.083	-0.54000	-0.27000	-0.00000	-0.00000
83			1.0010	-1.046	-0.20000	-0.10001	0.00000	0.00001
84			1.0056	1.170	-0.11000	-0.06999	0.00000	-0.00001
85		1	1.0150	2.540	-0.23999	0.02569	-0.00001	0.0
86			1.0053	1.121	-0.21000	-0.10000	0.00000	0.00000
87		1	1.0150	1.047	0.00000	0.02446	-0.00000	0.0
88			1.0178	5.578	-0.48000	-0.10002	0.00000	0.00002
89		1	1.0350	9.461	6.07001	0.55393	-0.00001	0.0
90		1	0.9850	3.726	-1.63000	-0.31008	-0.00000	0.0
91		1	0.9800	3.825	-0.10000	-0.32553	0.00000	0.0
92	1		1.0152	4.035	-0.64999	-0.01000	-0.00001	0.00000
93	2		1.0052	1.259	-0.12000	-0.07001	0.00000	0.00001
94	2		1.0045	-0.726	-0.29999	-0.16000	-0.00001	0.0
95	2		0.9926	-1.624	-0.42000	-0.30998	-0.00000	-0.00002
96	2		1.0013	-1.725	-0.38000	-0.15001	0.00000	0.00001
97	2		1.0133	-1.218	-0.15000	-0.08999	0.00000	-0.00001
98	2		1.0253	-1.693	-0.34000	-0.08001	0.00000	0.00001
99	2	1	1.0100	-2.082	-0.42000	-0.31259	0.00000	0.0
100	1	1	1.0300	-1.297	2.15001	1.06086	-0.00001	0.0
101			1.0102	0.131	-0.22000	-0.15001	0.00000	0.00001
102			1.0124	2.639	-0.05000	-0.02997	-0.00000	-0.00003
103			1.0144	-4.817	0.16999	0.23999	0.00001	0.00001
104			0.9812	-7.431	-0.38000	-0.32999	-0.00000	-0.00001
105		1	0.9800	-8.577	-0.31001	-0.15994	0.00001	0.0
106			0.9732	-8.778	-0.43000	-0.15999	-0.00000	-0.00001
107		1	0.9520	-11.335	-0.50000	-0.19716	-0.00000	0.0
108			0.9830	-9.770	-0.02000	-0.01000	-0.00000	-0.00000
109			0.9845	-10.219	-0.08000	-0.03001	0.00000	0.00001
110			0.9923	-11.065	-0.39000	-0.38001	-0.00000	0.00001
111		1	1.0350	-10.140	0.36000	0.47155	-0.00000	0.0
112		1	0.9750	-13.664	-0.68000	-0.01498	0.00000	0.0
113		1	0.9930	-14.035	-0.06000	0.15381	0.00000	0.0
114			0.9601	-13.557	-0.07999	-0.03002	-0.00001	0.00002
115			0.9600	-13.569	-0.22000	-0.06994	-0.00000	-0.00006
116	2	1	1.0050	-1.403	-1.83998	-3.01552	-0.00002	0.0
117			0.9784	-17.039	-0.20000	-0.08000	0.00000	0.00000
118			0.9604	-4.453	-0.33000	-0.15002	0.00000	0.00002

Input Data

1	0	0	0.9557	-0.2958	0.0
2	0	0	0.9745	-0.2872	0.0
3	0	0	0.9682	-0.2804	0.0
4	0	1	0.9980	-0.2152	0.0
5	0	0	0.9954	-0.2068	-0.4000
6	0	1	0.9900	-0.2554	0.0
7	0	0	0.9910	-0.2636	0.0
8	0	1	1.0150	-0.1172	0.0
9	0	0	1.0351	0.0110	0.0
10	0	1	1.0250	0.1468	0.0
11	0	0	0.9874	-0.2610	0.0
12	0	0	0.9945	-0.2707	0.0
13	0	0	0.9701	-0.2851	0.0
14	0	0	0.9869	-0.2832	0.0
15	0	1	0.9700	-0.2884	0.0
16	0	0	0.9862	-0.2757	0.0
17	0	0	0.9925	-0.2435	0.0
18	0	1	0.9730	-0.2832	0.0
19	0	0	0.9629	-0.2917	0.0
20	0	0	0.9554	-0.2774	0.0
21	0	0	0.9543	-0.2503	0.0
22	0	0	0.9636	-0.2061	0.0
23	0	0	0.9907	-0.1204	0.0
24	0	1	0.9920	-0.1241	0.0
25	0	1	1.0200	0.0058	0.0
26	0	1	1.0350	0.0357	0.0
27	0	1	0.9680	-0.2218	0.0
28	0	0	0.9616	-0.2511	0.0
29	0	0	0.9632	-0.2675	0.0
30	0	0	1.0141	-0.1529	0.0
31	0	1	0.9670	-0.2651	0.0
32	0	1	0.9630	-0.2304	0.0
33	0	0	0.9695	-0.2989	0.0
34	0	1	0.9840	-0.2878	0.1400
35	0	0	0.9799	-0.2955	0.0
36	0	1	0.9800	-0.2957	0.0
37	0	0	0.9877	-0.2790	-0.2500
38	0	0	1.0058	-0.1871	0.0
39	0	0	0.9688	-0.3393	0.0
40	0	1	0.9700	-0.3588	0.0
41	0	0	0.9668	-0.3667	0.0
42	0	1	0.9850	-0.3392	0.0
43	0	0	0.9841	-0.2905	0.0
44	0	0	1.0013	-0.2508	0.1000
45	0	0	1.0063	-0.2209	0.1000
46	0	1	1.0350	-0.1769	0.1000
47	0	0	1.0315	-0.1348	0.0
48	0	0	1.0350	-0.1480	0.1500
49	0	1	1.0350	-0.1294	0.0
50	0	0	1.0202	-0.1649	0.0
51	0	0	0.9974	-0.2095	0.0
52	0	0	0.9903	-0.2254	0.0
53	0	0	0.9867	-0.2416	0.0
54	0	1	1.0000	-0.2270	0.0
55	0	1	1.0000	-0.2321	0.0
56	0	1	1.0000	-0.2288	0.0
57	0	0	1.0055	-0.2083	0.0
58	0	0	0.9963	-0.2225	0.0
59	0	1	1.0350	-0.1593	0.0
60	0	0	1.0315	-0.0957	0.0
61	0	1	1.0350	-0.0813	0.0
62	0	0	1.0243	-0.0894	0.0

Output

63	0	0	1.0349	-0.1021	0.0
64	0	0	1.0366	-0.0724	0.0
65	0	1	1.0350	-0.0168	0.0
66	0	1	1.0350	-0.0147	0.0
67	0	0	1.0238	-0.0632	0.0
68	1	0	1.0172	-0.0182	0.0
69	0	1	1.0350	0.0276	0.0
70	0	0	0.9826	-0.0909	0.0
71	0	0	0.9862	-0.0988	0.0
72	0	1	0.9800	-0.1211	0.0
73	0	1	0.9910	-0.1027	0.0
74	0	0	0.9682	-0.1043	0.1200
75	0	0	0.9747	-0.0798	0.0
76	0	0	0.9579	-0.0583	0.0
77	1	0	1.0027	-0.0423	0.0
78	2	0	0.9999	-0.0473	0.0
79	2	0	1.0053	-0.0414	0.2000
80	2	2	1.0350	0.0	0.0
81	2	0	1.0293	-0.0123	0.0
82	1	0	0.9988	-0.0364	0.2000
83	0	0	1.0010	-0.0163	0.1000
84	0	0	1.0056	0.0204	0.0
85	0	1	1.0150	0.0443	0.0
86	0	0	1.0053	0.0196	0.0
87	0	1	1.0150	0.0183	0.0
88	0	0	1.0178	0.0974	0.0
89	0	1	1.0350	0.1651	0.0
90	0	1	0.9850	0.0650	0.0
91	0	1	0.9800	0.0668	0.0
92	1	0	1.0152	0.0704	0.0
93	2	0	1.0052	0.0220	0.0
94	2	0	1.0045	-0.0127	0.0
95	2	0	0.9926	-0.0283	0.0
96	2	0	1.0013	-0.0301	0.0
97	2	0	1.0133	-0.0213	0.0
98	2	0	1.0253	-0.0295	0.0
99	2	1	1.0100	-0.0363	0.0
100	1	1	1.0300	-0.0226	0.0
101	0	0	1.0102	0.0023	0.0
102	0	0	1.0124	0.0461	0.0
103	0	0	1.0144	-0.0841	0.0
104	0	0	0.9812	-0.1297	0.0
105	0	1	0.9800	-0.1497	0.2000
106	0	0	0.9732	-0.1532	0.0
107	0	1	0.9520	-0.1978	0.0600
108	0	0	0.9830	-0.1705	0.0
109	0	0	0.9845	-0.1784	0.0
110	0	0	0.9923	-0.1931	0.0600
111	0	1	1.0350	-0.1770	0.0
112	0	1	0.9750	-0.2385	0.0
113	0	1	0.9930	-0.2450	0.0
114	0	0	0.9601	-0.2366	0.0
115	0	0	0.9600	-0.2368	0.0
116	2	1	1.0050	-0.0245	0.0
117	0	0	0.9784	-0.2974	0.0
118	0	0	0.9604	-0.0777	0.0

316.32224	-10.69795	0.0	0.0	0.0	24.59744	-3.49223	0.0	0.0
-10.85768	123.27492	-10.44183	0.0	0.0	0.20416	36.92810	-3.71778	0.0
0.0	-10.48534	31.41103	-3.81208	0.0	0.0	-3.59323	9.86129	-0.5076
0.0	0.0	-3.93343	28.06381	-6.81845	0.0	0.0	-1.04629	9.0677
0.0	0.0	0.0	-6.57417	40.53008	0.0	0.0	0.0	-2.1741
-29.18988	0.0	0.0	0.0	0.0	359.38086	0.0	0.0	0.0
-0.03733	-41.69501	3.71778	0.0	0.0	0.00069	126.81151	-10.44183	0.0
0.0	3.59323	-12.05972	0.03299	0.0	0.0	-10.48534	34.69295	0.0043
0.0	0.0	0.0	-15.04833	-0.30284	0.0	0.0	0.0	56.9492

Output

68	1.4911	1.5745	1.0172	-0.0182
77	-0.4853	-0.0057	1.0027	-0.0423
82	0.4676	-0.0875	0.9988	-0.0364
92	2.0435	-0.3417	1.0152	0.0704
100	-2.2198	-0.4566	1.0300	-0.0226

Output

```
C
C     NEWT62.FORT
C     DIMENSIONED FOR 62 BUSSES.
C     MODIFIED TO HANDLE THE REDUCED SYSTEM.
C     READS PARAMETERS, BUS DATA AND ADMITTANCES ON TUBE 2.
C     READS CORRECTION MATRIX 'JCOR', POWER INJECTIONS, AND
C     BASE CASE VOLTAGES ON TUBE 10.
C
      COMPLEX WYE(62,62),V(62),CUR(62),P(62),S(62),CONJG,WHY,BP(5)
      DIMENSION Y(2,62,62),E(2,62),AYE(2,62),PWR(2,62),PQ(2,62)
      DIMENSION DP(62),DQ(62),VEE(62),ANG(62),YSH(62)
      DIMENSION GMW(62),GMV(62),GMVMIN(62),GMVMAX(62)
      DIMENSION JK(125),KB(180),LB(62),ITYPE(62),JTYPE(62)
      DIMENSION BPWR(2,5),BVEE(5),BANG(5),DEV(5),DBA(5),DBP(5),DEQ(5)
      EQUIVALENCE (WYE(1,1),Y(1,1,1)),(V(1),E(1,1)),(CUR(1),AYE(1,1))
      EQUIVALENCE (P(1),PWR(1,1)),(BP(1),BPWR(1,1)),(S(1),PQ(1,1))
      REAL JAKE(125,126),JCOR(10,10),LMW(62),LMV(62)
      DATA Y/7688*0.0/
C
C     READ PARAMETERS AND INITIALIZE.
C
      READ (2,202)NADM,NBUS,EPS,ITMAX
      ITER=0
      NLBD=0
      NVBD=0
C
C     READ BUS DATA
C
      DO 35 I=1,NBUS
      READ (2,201) IBUS,JTYPE(I),ITYPE(I),VEE(I),ANG(I),LMW(I),
     1LMV(I),GMW(I),GMV(I),GMVMIN(I),GMVMAX(I),YSH(I)
      IF (JTYPE(I).EQ.1.AND.ITYPE(I).EQ.0) NLBD=NLBD+1
      IF (JTYPE(I).EQ.1.AND.ITYPE(I).EQ.1) NVBD=NVBD+1
      KB(IBUS)=I
      LB(I)=IBUS
      ANG(I)=ANG(I)*3.14159/180.0
      E(1,I)=VEE(I)*COS(ANG(I))
      E(2,I)=VEE(I)*SIN(ANG(I))
      PQ(1,I)=GMW(I)-LMW(I)
      PQ(2,I)=GMV(I)-LMV(I)
   35 Y(2,I,I)=Y(2,I,I)+YSH(I)
      NHBD=NLBD+NVBD
      NJCOR=2*NLBD+NVBD
      NH=NBUS-1
C
C     READ CORRECTION MATRIX, POWER INJECTIONS AND BASE CASE VOLTAGES.
C
      DO 12 I=1,NJCOR
   12 READ (10,205) (JCOR(I,J),J=1,NJCOR)
      DO 14 I=1,NHBD
   14 READ (10,206) KBD,BP(I),BVEE(I),BANG(I)
C
C     BUILD ADMITTANCE MATRIX
C
      DO 20 I=1,NADM
      READ (2,200) IFR,ITO,WHY,CHG
      KBFR=KB(IFR)
      KBTO=KB(ITO)
      WYE(KBFR,KBTO)=-WHY
      WYE(KBTO,KBFR)=-WHY
      Y(2,KBFR,KBFR)=Y(2,KBFR,KBFR)+(CHG/2.0)
   20 Y(2,KBTO,KBTO)=Y(2,KBTO,KBTO)+(CHG/2.0)
      DO 30 I=1,NBUS
      DO 30 J=1,NBUS
   30 IF(I.NE.J) WYE(I,I)=WYE(I,I)-WYE(I,J)
```

Source Listing - Reduced System Load Flow

```
      GC TC 44
   40 DC 41 I=1.NH
      K=JK(I)
   41 ANG(K)=ANG(K)+JAKE(I.NJP)
      CO 42 I=NBUS.NJAC
      K=JK(I-NH)
   42 VEE(K)=VEE(K)*(1.0+JAKE(I.NJP))
      CO 43 I=1.NBUS
      E(1.I)=VEE(I)*CCS(ANG(I))
      E(2.I)=VEE(I)*SIN(ANG(I))
      PC(1.I)=GMW(I)-LMW(I)
   43 PQ(2.I)=GMV(I)-LMV(I)
      IF (ITER.GT.ITMAX) GC TO 110
C
C     CALCULATE CURRENTS AND PCWERS
C
   44 DC 45 I=1.NBUS
      DP(I)=0.0
      DG(I)=0.0
   45 CUR(I)=C.0
      DO 46 I=1.NHBD
      DBP(I)=0.
      DBQ(I)=0.
      IF (ITYPE(I).EQ.0) CEV(I)=BVEE(I)-VEE(I)
   46 DBA(I)=BANG(I)-ANG(I)
      DC 48 I=1.NHBD
      DO 48 J=1.NHBD
      DBP(I)=DBP(I)+DBA(J)*JCOR(I.J)
      IF (J.LE.NLBD) DBP(I)=DBP(I)+DBV(J)*JCOR(I.J+NHBD)
      IF (I.GT.NLBD) GC TC 48
      DBQ(I)=DBQ(I)+DBA(J)*JCOR(I+NHBD.J)
      IF (J.LE.NLBD) DBQ(I)=DBQ(I)+DBV(J)*JCOR(I+NHBD.J+NHBD)
   48 CONTINUE
      ICHK=0
      DC 50 I=1.NBUS
      IF (ITYPE(I).EQ.2) GC TO 50
      CC 52 J=1.NBUS
   52 CUR(I)=CUR(I)+YE(I.J)*V(J)
      P(I)=V(I)*CCNJG(CUR(I))
      DP(I)=PC(1.I)-PWR(1.I)
      IF (JTYPE(I).EC.1) DP(I)=DP(I)+DBP(I)+BPWR(1.I)
      IF (ABS(DP(I)).GT.EPS) ICHK=1
      IF (ITYPE(I).NE.0) GC TO 50
      DQ(I)=PC(2.I)-PWR(2.I)
      IF (JTYPE(I).EQ.1) DQ(I)=DQ(I)+DBQ(I)+BPWR(2.I)
      IF (ABS(DQ(I)).GT.EPS) ICHK=1
   50 CONTINUE
      IF(ICHK.EQ.1.AND.ITER.GT.0) GO TO 82
      IF(ICHK.EQ.1) GO TO 54
      DC 54 I=1.NBUS
      IF (ITYPE(I).EQ.2) GC TO 54
      GVAR=PWR(2.I)+LMV(I)
      IF(ITYPE(I).EQ.1.AND.GVAR.LT.GMVMIN(I)) GO TO 56
      IF(ITYPE(I).EQ.1.AND.GVAR.GT.GMVMAX(I)) GC TC 58
      GC TC 54
   56 GMV(I)=GMVMIN(I)
      GO TO 59
   58 GMV(I)=GMVMAX(I)
   59 ITYPE(I)=0
      ICHK=1
      PG(2.I)=GMV(I)-LMV(I)
      DQ(I)=PQ(2.I)-PWR(2.I)
   54 CCNTINUE
      IF (ICHK.EQ.0) GC TC 110
C
C     CCUNT JACCBIAN
```

Source Listing - Reduced System Load Flow

```
C
      NLD=0
      NVC=0
      DO 60 I=1,NBUS
      J=ITYPE(I)-1
      IF (J) 61,62,60
   61 NLD=NLD+1
      GO TO 60
   62 NVC=NVC+1
   60 CONTINUE
      NJAC=2*NLD+NVC
      NJP=NJAC+1
C
C     REORDER BUSSES
C
      K=0
      L=NLD
      DO 70 I=1,NBUS
      J=ITYPE(I)-1
      IF (J) 71,72,70
   71 K=K+1
      JK(K)=I
      GO TO 70
   72 L=L+1
      JK(L)=I
   70 CONTINUE
C
C     BUILD JACOBIAN
C
   82 ITER=ITER+1
      DO 80 I=1,NH
      K=JK(I)
      DO 80 J=1,NH
      M=JK(J)
      IF (K.EQ.M) GO TO 85
      A=E(1,M)*Y(1,K,M)-E(2,M)*Y(2,K,M)
      B=E(2,M)*Y(1,K,M)+E(1,M)*Y(2,K,M)
      JAKE(I,J)=(A*E(2,K))-(B*E(1,K))
      IF (I.LE.NLD.AND.J.LE.NLD) JAKE(I+NH,J+NH)=JAKE(I,J)
      GO TO 80
   85 JAKE(I,J)=-PWR(2,K)-(Y(2,K,K)*VEE(K)*VEE(K))
      IF(I.LE.NLD.AND.J.LE.NLD)JAKE(I+NH,J+NH)=JAKE(I,J)+2.*PWR(2,K)
   80 CONTINUE
      DO 90 I=1,NH
      K=JK(I)
      DO 90 J=NBUS,NJAC
      M=JK(J-NH)
      IF (K.EQ.M) GO TO 95
      A=E(1,M)*Y(1,K,M)-E(2,M)*Y(2,K,M)
      B=E(2,M)*Y(1,K,M)+E(1,M)*Y(2,K,M)
      JAKE(I,J)=(A*E(1,K))+(B*E(2,K))
      JAKE(I+NH,J-NH)=-JAKE(I,J)
      GO TO 90
   95 JAKE(I,J)=PWR(1,K)+(Y(1,K,K)*VEE(K)*VEE(K))
      JAKE(J,I)=PWR(1,K)-(Y(1,K,K)*VEE(K)*VEE(K))
   90 CONTINUE
C
C     CORRECT JACOBIAN
C
      DO 96 I=1,NJCOR
      K=I
      IF (I.GT.NLBD) K=NLD-NLBD+I
      IF (I.GT.NHBD) K=NH-NHBD+I
      DO 96 J=1,NJCOR
      M=J
      IF (J.GT.NLBD) M=NLD-NLBD+J
```

```
      IF (J.GT.NHBD) M=NH-NHBD+J
   96 JAKE(K,M)=JAKE(K,M)+JCCP(I,J)
C
C     BUILD AUGMENT VECTOR
C
      DO 100 I=1,NH
      K=JK(I)
  100 JAKE(I,NJP)=DP(K)
      DO 105 I=NBUS,NJAC
      K=JK(I-NH)
  105 JAKE(I,NJP)=DQ(K)
      CALL SOLVE (NJAC,JAKE)
      GO TO 40
  110 WRITE (6,203) ITER
      DO 47 I=1,NBUS
      ARC=ANG(I)*180.0/3.14159
   47 WRITE (6,204) LB(I),JTYPE(I),ITYPE(I),VEE(I),ARC,P(I),DP(I),DQ(I
      STOP
  200 FORMAT (2I5,3F10.5)
  201 FORMAT (I4,2I2,9F8.5)
  202 FORMAT (2I5,10X,E10.4,I5)
  203 FORMAT (/1X,'ITERATION NUMBER',I4/4X,'BUS',5X,'BUS',5X,'BUS',5X
     1,'BUS',10X,'POWER',11X,'MISMATCH'/2X,'NUMBER',2X,'TYPES',
     22X,'VOLTAGE',2X,'ANGLE',4X,2('REAL',3X,'REACTIVE',3X))
  204 FORMAT (1X,I5,3X,2I3,2X,F8.4,F8.3,4F9.5)
  205 FORMAT (12F10.5)
  206 FORMAT (I10,4F12.4)
      END
      SUBROUTINE SOLVE (N,A)
      DIMENSION A(125,126)
      MAX=N+1
      DO 12 K=1,N
      KP=MAX+1-K
      DO 10 J=1,KP
      JP=MAX+1-J
   10 A(K,JP)=A(K,JP)/A(K,K)
      DO 12 I=1,N
      IF (I.EQ.K) GO TO 12
      DO 12 J=1,KP
      JP=MAX+1-J
      A(I,JP)=A(I,JP)-A(I,K)*A(K,JP)
   12 CONTINUE
      RETURN
      END
```

Source Listing - Reduced System Load Flow

APPENDIX C

This appendix contains the FORTRAN source listing, a description of the input and an output list for the final segment of the program. The input is read from two separate data files. The first file contains the Jacobian correction matrix and boundary bus conditions produced by the previous segment and described in Appendix B. The second file is the same format as the input to the first segment, with three features added. The first line is exactly the same format as described in Appendix A. The next four lines contain coefficients used for different load types, such as

1.0	1.96	0.501	1.77	1.0	2.40	11.	55.6
(A)	(B)	(C)	(D)	(E)	(F)	(G)	(H)

where (A) and (E) are constant multipliers for P and Q respectively,

(B) and (F) are coefficients of ΔV for P and Q,

(C) and (G) are coefficients of $(\Delta V)^2$ for P and Q,

and (D) and (H) are coefficients of $(\Delta V)^3$ for P and Q.

In the section of bus data two additional bus type codes appear:

100	111	1.030	. . .
(A)	(B)	(C)	(D)

where (A) is the bus number

(B) is the load type code; 1 for general, 2 for residential, 3 for commercial, 4 for industrial.

(C) is the reduction code. In this segment only the presence or absence of the 1, signifying a boundary node, is pertinent.

(D) is the bus type code.

The remainder of the line is the same as in the other segments, as in the entire section on line data.

The output is self-explanatory.

```
C
C     NEWVAR62.FORT
C     DIMENSIONED FOR 62 BUSSES.
C     TREATS ALL LOADS AS VOLTAGE VARIABLE.
C     MODIFIED TO HANDLE THE REDUCED SYSTEM.
C     READS PARAMETERS. BUS DATA AND ADMITTANCES ON TUBE 2.
C     READS CORRECTION MATRIX 'JCCR'. POWER INJECTIONS. AND
C     BASE CASE VOLTAGES ON TUBE 10.
C
      COMPLEX WYE(62.62).V(62).CUR(62).P(62).S(62).CONJG.WHY.BP(5)
      DIMENSION Y(2.62.62).E(2.62).AYE(2.62).PWR(2.62).FC(2.62)
      DIMENSION DP(62).DQ(62).VEE(62).ANG(62).YSH(62).AA(4.10).BB(4.10)
      DIMENSION GMW(62).GMV(62).GMVMIN(62).GMVMAX(62).VMW(62).VMV(62)
      DIMENSION JK(125).KB(180).LB(62).ITYPE(62).JTYPE(62).LTYPE(62)
      DIMENSION BPWR(2.5).EVEE(5).EANG(5).DEV(5).DBA(5).DBP(5).DBQ(5)
      EQUIVALENCE (WYE(1.1).Y(1.1.1)).(V(1).E(1.1)).(CUR(1).AYE(1.1))
      EQUIVALENCE (P(1).PWR(1.1)).(BP(1).BPWR(1.1)).(S(1).FC(1.1))
      REAL JAKE(125.126).JCCR(10.10).LMW(62).LMV(62)
C
C     READ PARAMETERS AND INITIALIZE.
C
      READ (2.202)NADM.NBUS.EPS.ITMAX
      DO 5 J=1.4
    5 READ (2.207) (AA(I.J).I=1.4).(BB(I.J).I=1.4)
      DO 10 I=1.NBUS
      DO 10 J=1.NBUS
   10 WYE(I.J)=0.
      ITER=0
      NLBD=0
      NVBD=0
C
C     READ BUS DATA
C
      DO 25 I=1.NBUS
      READ (2.201) IBUS.LTYPE(I).JTYPE(I).ITYPE(I).VEE(I).ANG(I).
     1LMW(I).LMV(I).GMW(I).GMV(I).GMVMIN(I).GMVMAX(I).YSH(I)
      IF (JTYPE(I).EQ.1.AND.ITYPE(I).EQ.0) NLBD=NLBD+1
      IF (JTYPE(I).EQ.1.AND.ITYPE(I).EQ.1) NVBD=NVBD+1
      KB(IBUS)=I
      LB(I)=IBUS
      ANG(I)=ANG(I)*3.14159/180.0
   25 Y(2.I.I)=Y(2.I.I)+YSH(I)
      NRED=NLBD+NVBD
      NJCCR=2*NLBD+NVBD
      NM=NBUS-1
C
C     READ CORRECTION MATRIX. POWER INJECTIONS AND BASE CASE VOLTAGES.
C
      DO 12 I=1.NJCCR
   12 READ (10.205) (JCCR(I.J).J=1.NJCCR)
      DO 14 I=1.NRED
   14 READ (10.206) KBD.BP(I).EVEE(I).EANG(I)
C
C     BUILD ADMITTANCE MATRIX
C
      DO 20 I=1.NADM
      READ (2.200) IFR.ITO.WHY.CHC
      KBFR=KB(IFR)
```

Source Listing - Combined Load Flow

```
      KBTO=KB(ITO)
      WYE(KBFR,KBTO)=-WHY
      WYE(KBTO,KBFR)=-WHY
      Y(2,KBFR,KBFR)=Y(2,KBFR,KBFR)+(CHG/2.0)
   20 Y(2,KBTO,KBTO)=Y(2,KBTO,KBTO)+(CHG/2.0)
      DO 30 I=1,NBUS
      DO 30 J=1,NBUS
   30 IF(I.NE.J) WYE(I,I)=WYE(I,I)-WYE(I,J)
      GO TO 44
   40 DO 41 I=1,NH
      K=JK(I)
   41 ANG(K)=ANG(K)+JAKE(I,NJP)
      DO 42 I=NBUS,NJAC
      K=JK(I-NH)
   42 VEE(K)=VEE(K)+JAKE(I,NJP)
   44 DO 43 I=1,NBUS
      L=LTYPE(I)
      E(1,I)=VEE(I)*COS(ANG(I))
      E(2,I)=VEE(I)*SIN(ANG(I))
      DV=VEE(I)-1.0
      AAA=(AA(2,L)*DV+AA(3,L)*DV*DV+AA(4,L)*DV*DV*DV)/AA(1,L)
      VMW(I)=LMW(I)*(1.0+AAA)
      BBB=(BB(2,L)*DV+BB(3,L)*DV*DV+BB(4,L)*DV*DV*DV)/BB(1,L)
      VMV(I)=LMV(I)*(1.0+BBB)
      PC(1,I)=GMW(I)-VMW(I)
   43 PC(2,I)=GMV(I)-VMV(I)
      IF (ITER.GT.ITMAX) GO TO 110
C
C     CALCULATE CURRENTS AND POWERS
C
      DO 45 I=1,NBUS
      DP(I)=0.0
      DQ(I)=0.0
   45 CUR(I)=0.0
      DO 46 I=1,NHED
      DEP(I)=0.
      DEQ(I)=0.
      IF (ITYPE(I).EQ.0) DEV(I)=EVEE(I)-VEE(I)
   46 DEA(I)=EANG(I)-ANG(I)
      DO 48 I=1,NHED
      DO 48 J=1,NHED
      DEP(I)=DEP(I)+DEA(J)*JCOR(I,J)
      IF (J.LE.NLED) DEP(I)=DBP(I)+DBV(J)*JCOR(I,J+NHED)
      IF (I.GT.NLED) GO TO 48
      DEQ(I)=DEQ(I)+DEA(J)*JCOR(I+NHED,J)
      IF (J.LE.NLED) DEQ(I)=DBQ(I)+DBV(J)*JCOR(I+NHED,J+NHED)
   48 CONTINUE
      ICHK=0
      DO 50 I=1,NBUS
      IF (ITYPE(I).EQ.2) GO TO 50
      DO 52 J=1,NBUS
   52 CUR(I)=CUR(I)+WYE(I,J)*V(J)
      P(I)=V(I)*CONJG(CUR(I))
      DP(I)=PC(1,I)-PWR(1,I)
      IF (JTYPE(I).EQ.1) DP(I)=DP(I)+DBP(I)+BFWR(1,I)
      IF (ABS(DP(I)).GT.EPS) ICHK=1
      IF (ITYPE(I).NE.0) GO TO 50
      DQ(I)=PC(2,I)-PWR(2,I)
```

```
      IF (JTYPE(I).EQ.1) DQ(I)=DQ(I)+DBQ(I)+EPWR(2,I)
      IF (ABS(DQ(I)).GT.EPS) ICHK=1
   50 CONTINUE
      IF(ICHK.EQ.1.AND.ITER.GT.0) GO TO 82
      IF(ICHK.EQ.1) GO TO 54
      DO 54 I=1,NBUS
      IF (ITYPE(I).EQ.2) GO TO 54
      GVAR=PWR(2,I)+LMV(I)
      IF(ITYPE(I).EQ.1.AND.GVAR.LT.GMVMIN(I)) GO TO 56
      IF(ITYPE(I).EQ.1.AND.GVAR.GT.GMVMAX(I)) GO TO 58
      GO TO 54
   56 GMV(I)=GMVMIN(I)
      GO TO 55
   58 GMV(I)=GMVMAX(I)
   55 ITYPE(I)=0
      ICHK=1
      PC(2,I)=GMV(I)-LMV(I)
      DQ(I)=PC(2,I)-PWR(2,I)
   54 CONTINUE
      IF (ICHK.EQ.0) GO TO 110
C
C     COUNT JACOBIAN
C
      NLD=0
      NVC=0
      DO 60 I=1,NBUS
      J=ITYPE(I)-1
      IF (J) 61,62,60
   61 NLD=NLD+1
      GO TO 60
   62 NVC=NVC+1
   60 CONTINUE
      NJAC=2*NLD+NVC
      NJP=NJAC+1
C
C     REORDER BUSSES
C
      K=0
      L=NLD
      DO 70 I=1,NBUS
      J=ITYPE(I)-1
      IF (J) 71,72,70
   71 K=K+1
      JK(K)=I
      GO TO 70
   72 L=L+1
      JK(L)=I
   70 CONTINUE
C
C     BUILD JACOBIAN
C
   82 ITER=ITER+1
      DO 80 I=1,NH
      K=JK(I)
      DO 80 J=1,NH
      M=JK(J)
      IF (K.EQ.M) GO TO 85
      A=E(1,M)*Y(1,K,M)-E(2,M)*Y(2,K,M)
```

Source Listing - Combined Load Flow

```
      B=E(2,M)*Y(1,K,M)+E(1,M)*Y(2,K,M)
      JAKE(I,J)=(A*E(2,K))-(B*E(1,K))
      IF (I.LE.NLD.AND.J.LE.NLD) JAKE(I+NH,J+NH)=JAKE(I,J)
      GC TO 80
   85 JAKE(I,J)=-FWR(2,K)-(Y(2,K,K)*VEE(K)*VEE(K))
      IF(I.LE.NLD.AND.J.LE.NLD)JAKE(I+NH,J+NH)=JAKE(I,J)+2.*FWR(2,K)
   8C CCNTINUE
      DC 9C I=1,NH
      K=JK(I)
      DC 90 J=NELS,NJAC
      M=JK(J-NH)
      IF (K.EC.M) GC TC 95
      A=E(1,M)*Y(1,K,M)-E(2,M)*Y(2,K,M)
      B=E(2,M)*Y(1,K,M)+E(1,M)*Y(2,K,M)
      JAKE(I,J)=(A*E(1,K))+(B*E(2,K))
      JAKE(I+NH,J-NH)=-JAKE(I,J)
      GC TO 90
   95 JAKE(I,J)=FWR(1,K)+(Y(1,K,K)*VEE(K)*VEE(K))
      JAKE(J,I)=FWR(1,K)-(Y(1,K,K)*VEE(K)*VEE(K))
   9C CCNTINUE
C
C     CCRRECT JACCBIAN
C
      DC 96 I=1,NJCCR
      K=I
      IF (I.GT.NLEC) K=NLD-NLEC+I
      IF (I.GT.NHBC) K=NH-NHBC+I
      DC 96 J=1,NJCCR
      M=J
      IF (J.GT.NLBC) M=NLD-NLBD+J
      IF (J.GT.NHBC) M=NH-NHBC+J
   96 JAKE(K,M)=JAKE(K,M)+JCCR(I,J)
C
C     BUILC ALGMENT VECTCR
C
      DC 100 I=1,NH
      K=JK(I)
  1CC JAKE(I,NJP)=CP(K)
      DC 105 I=NELS,NJAC
      K=JK(I-NH)
  105 JAKE(I,NJP)=CQ(K)
      CALL SCLVE (NJAC,JAKE)
      GC TC 4C
  11C WRITE (6,203) ITER
      DC 47 I=1,NELS
      ARC=ANG(I)*18C.0/3.14159
   47 WRITE (6,204) LB(I),LTYPE(I),JTYPE(I),ITYPE(I),VEE(I),ARC,
     1  F(I),CF(I),CC(I)
      STCP
  20C FCRMAT (2I5,3F1C.5)
  2C1 FCRMAT (I4,I2,2I1,9F8.5)
  2C2 FCRMAT (2I5,10X,E1C.4,I5)
  203 FCRMAT (/,1X,'ITERATICN NUMBER',I4,/,4X,'BUS',5X,'BUS',5X,'BUS',5)
     1,'BUS',1CX,'PCWER',11X,'MISMATCH',/,2X,'NUMBER',3X,'TYPES',
     22X,'VOLTAGE',2X,'ANGLE',4X,2('REAL',3X,'REACTIVE',3X))
  2C4 FCRMAT (1X,I5,4X,3I2,1X,F8.4,F8.3,4F9.5)
  2C5 FCRMAT (12F1C.5)
  2C6 FCRMAT (I10,4F12.4)
```

```
      SUBROUTINE SOLVE (N,A)
      DIMENSION A(125,126)
      MAX=N+1
      DO 12 K=1,N
      KP=MAX+1-K
      DO 10 J=1,KP
      JP=MAX+1-J
   10 A(K,JP)=A(K,JP)/A(K,K)
      DO 12 I=1,N
      IF (I.EQ.K) GO TO 12
      DO 12 J=1,KP
      JP=MAX+1-J
      A(I,JP)=A(I,JP)-A(I,K)*A(K,JP)
   12 CONTINUE
      RETURN
      END
```

Source Listing - Combined Load Flow

```
   65   56               1.0E-03   10
1.0                                           1.0
1.0        1.96       0.501       1.77        1.0         2.40       11.6       55.6
1.0        1.02      -4.23       22.7         1.0        -3.21      -57.9      -192.
1.0        1.11      -3.14       20.2         1.0        -0.0525    -25.5      -107.
  98  11    1.000   0.0    0.0      0.0          0.0      0.0     0.0      0.0     0.0
  77  11    1.000   0.0    0.610    0.280        0.0      0.0    -0.200    0.700   0.0
  52  11    1.000   0.0    0.540    0.270        0.0      0.0     0.0      0.0     0.20
  92  11    1.000   0.0    0.650    0.100        0.0      0.0    -0.030    0.090   0.0
 100 111    1.030   0.0    0.270    0.180        2.520    0.0    -0.500    1.550   0.0
  78  12    1.000   0.0    0.710    0.260        0.0      0.0     0.0      0.0     0.0
  79  12    1.000   0.0    0.250    0.220        0.0      0.0     0.0      0.0     0.20
  80 122    1.035   0.0    1.200    0.260        4.770    0.0    -1.650    2.800   0.0
  81  12    1.000   0.0    0.0      0.0          0.0      0.0     0.0      0.0     0.0
  93  12    1.000   0.0    0.120    0.070        0.0      0.0     0.0      0.0     0.0
  94  12    1.000   0.0    0.300    0.160        0.0      0.0     0.0      0.0     0.0
  95  12    1.000   0.0    0.420    0.210        0.0      0.0     0.0      0.0     0.0
  96  12    1.000   0.0    0.380    0.150        0.0      0.0     0.0      0.0     0.0
  97  12    1.000   0.0    0.007    0.002        0.0      0.0     0.0      0.0     0.0
  98  12    1.000   0.0    0.240    0.080        0.0      0.0     0.0      0.0     0.0
  99 121    1.010   0.0    0.0      0.0         -0.420    0.0    -1.000    1.000   0.0
 110 121    1.005   0.0    0.0      0.0         -1.840    0.0   -10.000   10.000   0.0
  51 3      1.000   0.0    0.00388 0.00291
  53 3      1.000   0.0    0.00380 0.00285
  54 3      1.000   0.0    0.00410 0.00307
  62 2      1.000   0.0    0.00205 0.00154
  63 2      1.000   0.0    0.00128 0.00096
  64 3      1.000   0.0    0.00773 0.00580
  66 2      1.000   0.0    0.00358 0.00268
  67 3      1.000   0.0    0.0     0.00270
  86 2      1.000   0.0    0.00110 0.00082
  83 2      1.000   0.0    0.00486 0.00365
  84 2      1.000   0.0    0.01498 0.01123
  85 2      1.000   0.0    0.00389 0.00292
  87 2      1.000   0.0    0.00086 0.00065
  88 4      1.000   0.0    0.01832 0.01374
  89 2      1.000   0.0    0.00205 0.00154
  41 3      1.000   0.0    0.00074 0.00055
  45 2      1.000   0.0    0.00094 0.00070
  46 2      1.000   0.0    0.00155 0.00116
  42 2      1.000   0.0    0.00446 0.00335
  47 2      1.000   0.0    0.00178 0.00133
  48 2      1.000   0.0    0.00084 0.00063
  43 3      1.000   0.0    0.00342 0.00256
  44 2      1.000   0.0    0.00120 0.00090
   1 2      1.000   0.0    0.01118 0.00835
   5 2      1.000   0.0    0.00466 0.00350
  15 2      1.000   0.0    0.00080 0.00060
   6 2      1.000   0.0    0.00768 0.00576
   8 2      1.000   0.0    0.00760 0.00570
  10 2      1.000   0.0    0.00582 0.00437
   2 2      1.000   0.0    0.00152 0.00114
   3 2      1.000   0.0    0.00032 0.00024
  31 2      1.000   0.0    0.00080 0.00060
   4 2      1.000   0.0    0.00052 0.00039
  21 2      1.000   0.0    0.00144 0.00108
 101 2      1.000   0.0    0.00147 0.00110
 102 2      1.000   0.0    0.00507 0.00380
 103 2      1.000   0.0    0.00433 0.00325
 104 3      1.000   0.0    0.00580 0.00435
 105 2      1.000   0.0    0.00089 0.00067
 121 2      1.000   0.0    0.00348 0.00261
 111 3      1.000   0.0    0.00810 0.00607
```

Input Data

62	91	4.12696	-49.15581	0.80800
68	116	15.64506	-248.60162	0.18400
77	78	22.09615	-74.05199	0.01260
77	80	8.90905	-27.30460	0.07000
77	82	3.65011	-10.44814	0.08180
78	79	6.64664	-39.07002	0.00640
79	80	3.00028	-12.52572	0.01860
80	96	1.03514	-5.29203	0.04940
80	97	2.02021	-10.31082	0.02540
80	98	1.94596	-8.83043	0.02860
80	99	1.02029	-4.62951	0.05400
81	80	0.0	-27.02702	0.0
82	96	5.27440	-17.25575	0.05440
92	93	3.28283	-10.79337	0.02180
92	94	1.76235	-5.79230	0.04060
92	100	0.71034	-3.23380	0.07720
93	94	3.20837	-12.50101	0.01880
94	95	6.41462	-21.09048	0.01100
94	96	3.25067	-10.50124	0.02300
94	100	4.83585	-15.75728	0.06040
95	96	5.20627	-14.65398	0.01480
96	97	2.12752	-10.88355	0.02400
98	100	1.18095	-5.32467	0.04760
99	100	2.59602	-11.72536	0.02160
97	51	0.50614	-1.15247	
51	53	1.01228	-2.30495	
53	54	0.62096	-0.81306	
97	62	0.28515	-0.65828	
62	63	0.11329	-0.04076	
62	64	0.85109	-1.37409	
64	66	1.09425	-1.76669	
66	67	0.12723	-0.07039	
97	86	0.26244	-0.59758	
86	83	0.60737	-1.28297	
83	84	0.54508	-1.24113	
84	85	0.26529	-0.22662	
97	87	0.82699	-0.45748	
87	88	0.45832	-0.16491	
88	89	0.14372	-0.05171	
97	41	1.01228	-2.30495	
41	45	0.26020	-0.09263	
45	46	0.21136	-0.05126	
41	42	0.64418	-1.46679	
42	47	0.54713	-0.88234	
47	48	0.38917	-0.21529	
42	43	0.24483	-0.20914	
43	44	0.15445	0.13193	
97	1	0.47240	-1.07564	
1	5	0.94396	-2.14937	
5	15	0.74537	-1.69719	
5	6	0.19897	-0.16996	
6	8	0.29478	-0.25180	
8	10	0.17301	-0.14780	
97	2	1.14473	-0.97787	
2	21	3.38656	-4.43421	
2	3	5.34207	-4.56339	
3	31	3.64232	-3.11140	
3	4	10.67465	-9.11868	
97	101	0.75983	-2.82352	
101	111	2.27903	-8.46881	
101	102	5.72264	-4.88935	
102	121	1.85033	-1.02358	
102	103	4.06388	-5.32105	
103	104	3.38656	-4.43421	
104	105	3.48396	-2.97612	

Input Data

ITERATION NUMBER 5

BUS NUMBER	BUS TYPES	BUS VOLTAGE	BUS ANGLE	POWER REAL	POWER REACTIVE	MISMATCH REAL	MISMATCH REACTIVE
68	1 1 0	1.0167	-1.041	1.49024	1.76157	0.00002	0.00021
77	1 1 0	1.0032	-2.433	-1.09468	-0.36117	-0.00002	-0.00002
82	1 1 0	0.9985	-2.080	-0.07013	-0.23900	-0.00000	-0.00000
92	1 1 0	1.0130	4.076	1.39245	-0.30279	0.00001	0.00004
100	1 1 1	1.0300	-1.298	-0.06770	0.72527	0.00000	0.0
78	1 2 0	1.0004	-2.716	-0.71001	-0.26006	0.00001	0.00006
79	1 2 0	1.0056	-2.375	-0.29000	-0.32003	0.00000	0.00003
80	1 2 2	1.0350	0.0	0.0	0.0	0.0	0.0
81	1 2 0	1.0290	-0.704	-0.00000	-0.00005	0.00000	0.00005
93	1 2 0	1.0027	1.281	-0.12000	-0.07000	-0.00000	0.00000
94	1 2 0	1.0035	-0.718	-0.30000	-0.16003	-0.00000	0.00003
95	1 2 0	0.9917	-1.620	-0.42000	-0.31003	0.00000	0.00003
96	1 2 0	1.0004	-1.724	-0.26000	-0.14998	-0.00000	-0.00002
97	1 2 0	1.0112	-1.230	-0.00696	-0.00189	-0.00004	-0.00011
98	1 2 0	1.0253	-1.694	-0.34000	-0.08001	-0.00000	0.00001
99	1 2 1	1.0100	-2.082	-0.42000	-0.31258	0.00000	0.0
116	1 2 1	1.0050	-1.406	-1.83999	-2.88768	-0.00001	0.0
51	3 0 0	1.0010	-1.552	-0.00289	-0.00290	0.00000	0.00000
53	3 0 0	0.9975	-1.662	-0.00379	-0.00287	-0.00000	-0.00000
54	3 0 0	0.9926	-1.737	-0.00407	-0.00313	0.00000	-0.00000
62	2 0 0	0.9853	-1.833	-0.00159	-0.00149	-0.00000	0.00000
63	2 0 0	0.9730	-1.614	-0.00121	-0.00091	0.00000	0.00000
64	3 0 0	0.9756	-1.560	-0.00752	-0.00593	0.00000	0.00000
66	2 0 0	0.9725	-1.564	-0.00239	-0.00253	0.00000	0.00000
67	3 0 0	0.9621	-0.975	-0.00000	-0.00269	0.00000	0.00000
86	2 0 0	0.9715	-2.444	-0.00104	-0.00077	-0.00000	-0.00000
83	2 0 0	0.9553	-2.574	-0.00444	-0.00333	0.00000	0.00000
84	2 0 0	0.9410	-3.454	-0.01328	-0.00998	0.00001	0.00001
85	2 0 0	0.9280	-3.502	-0.00325	-0.00253	0.00000	0.00000
87	2 0 0	0.9845	-1.065	-0.00083	-0.00063	-0.00001	-0.00000
88	4 0 0	0.9263	-0.258	-0.01672	-0.01204	0.00002	0.00002
89	2 0 0	0.9215	0.026	-0.00174	-0.00132	0.00000	0.00000
41	3 0 0	1.0048	-1.423	-0.00074	-0.00054	-0.00000	-0.00000
45	2 0 0	0.9941	-1.250	-0.00093	-0.00069	0.00000	0.00000
46	2 0 0	0.9861	-1.052	-0.00151	-0.00112	0.00000	0.00000
42	2 0 0	0.9968	-1.686	-0.00442	-0.00333	0.00000	0.00000
47	2 0 0	0.9939	-1.752	-0.00176	-0.00131	0.00000	0.00000
48	2 0 0	0.9916	-1.733	-0.00083	-0.00062	0.00000	0.00000
43	3 0 0	0.9788	-1.736	-0.00334	-0.00263	0.00000	0.00000
44	2 0 0	0.9772	-1.325	-0.00115	-0.00086	0.00000	0.00000
1	2 0 0	0.9781	-2.255	-0.01074	-0.00794	0.00004	-0.00005
5	2 0 0	0.9667	-2.621	-0.00401	-0.00307	-0.00034	-0.00019
15	2 0 0	0.9662	-2.636	-0.00076	-0.00057	0.00001	0.00001
6	2 0 0	0.8738	-2.918	-0.00594	-0.00452	0.00013	0.00009
8	2 0 0	0.8331	-3.065	-0.00585	-0.00448	0.00070	0.00069
10	2 0 0	0.8026	-3.189	-0.00445	-0.00327	0.00084	0.00096
2	2 0 0	1.0074	-1.243	-0.00154	-0.00116	-0.00000	-0.00000
3	2 0 0	1.0071	-1.244	-0.00031	-0.00023	-0.00001	-0.00001
31	2 0 0	1.0069	-1.245	-0.00082	-0.00062	0.00001	0.00000
4	2 0 0	1.0071	-1.244	-0.00053	-0.00040	0.00000	0.00000
21	2 0 0	1.0071	-1.248	-0.00146	-0.00110	0.00000	0.00000
101	2 0 0	1.0014	-1.664	-0.00146	-0.00108	-0.00001	-0.00003
102	2 0 0	0.9982	-1.675	-0.00506	-0.00379	0.00001	0.00001
103	2 0 0	0.9962	-1.707	-0.00430	-0.00322	0.00000	0.00000
104	3 0 0	0.9947	-1.730	-0.00577	-0.00441	-0.00000	-0.00000
105	2 0 0	0.9945	-1.730	-0.00088	-0.00066	0.00000	0.00000
121	2 0 0	0.9961	-1.659	-0.00346	-0.00259	0.00000	0.00000
111	3 0 0	1.0005	-1.705	-0.00811	-0.00607	0.00001	0.00001

Output

APPENDIX D

This appendix contains the output list for the final segment of the first test of the load flow program.

ITERATION NUMBER 4

BUS NUMBER	BUS TYPES	BUS VOLTAGE	BUS ANGLE	POWER REAL	POWER REACTIVE	MISMATCH REAL	MISMATCH REACTIVE
44	1 1 0	0.9997	-15.961	-0.32685	-0.09044	-0.00001	-0.00003
49	1 1 1	1.0350	-9.029	0.50060	-0.00160	-0.00014	0.0
69	1 1 2	1.0350	0.0	0.0	0.0	0.0	0.0
45	1 2 0	1.0056	-14.256	-0.53002	-0.21933	0.00002	-0.00067
46	1 2 1	1.0350	-11.747	-0.08994	0.13265	-0.00006	0.0
47	1 2 0	1.0315	-9.333	-0.34006	0.00073	0.00006	-0.00073
48	1 2 0	1.0353	-10.110	-0.03015	-0.00932	0.00015	-0.00068
1	2 0 0	1.0122	-11.139	-0.00006	-0.00013	0.00006	0.00013
11	2 0 0	1.0083	-11.324	-0.00058	-0.00027	0.00000	-0.00001
12	3 0 0	1.0040	-11.339	-0.00676	-0.00324	-0.00002	0.00001
13	2 0 0	1.0057	-11.451	-0.00322	-0.00156	-0.00001	-0.00001
14	2 0 0	1.0056	-11.452	-0.00022	-0.00011	-0.00000	-0.00000
15	2 0 0	1.0051	-11.448	-0.00127	-0.00062	-0.00001	-0.00000
16	2 0 0	1.0051	-11.448	-0.00072	-0.00035	-0.00000	-0.00000
17	2 0 0	1.0042	-11.523	-0.00067	-0.00033	-0.00001	-0.00000
18	2 0 0	1.0035	-11.557	-0.00522	-0.00255	-0.00001	0.00000
19	2 0 0	1.0032	-11.565	-0.00247	-0.00120	-0.00001	-0.00001
2	2 0 0	1.0073	-11.353	0.00001	-0.00001	-0.00001	0.00001
21	2 0 0	0.9941	-12.794	-0.00293	-0.00147	0.00013	0.00012
22	2 0 0	0.9539	-14.261	-0.00249	-0.00121	-0.00000	0.00000
23	2 0 0	0.9330	-14.327	-0.00207	-0.00101	-0.00008	-0.00004
24	2 0 0	0.9322	-14.323	-0.00007	-0.00003	-0.00000	-0.00000
25	2 0 0	0.9419	-14.734	-0.00444	-0.00212	-0.00011	-0.00009
27	2 0 0	0.9547	-14.201	-0.00626	-0.00303	-0.00009	-0.00004
28	2 0 0	0.9398	-14.760	-0.00573	-0.00274	-0.00016	-0.00012
3	2 0 0	0.9716	-12.861	-0.00108	-0.00052	0.00004	0.00002
31	2 0 0	0.9504	-12.930	-0.00208	-0.00100	-0.00006	-0.00003
32	2 0 0	0.9383	-12.962	-0.00385	-0.00188	-0.00012	-0.00005
33	2 0 0	0.9294	-12.990	-0.00201	-0.00098	-0.00010	-0.00005
4	2 0 0	1.0032	-11.563	-0.00131	-0.00066	0.00004	0.00005
5	4 0 0	0.9951	-11.852	-0.00894	-0.00437	-0.00001	0.00001
6	2 0 0	0.9898	-12.045	-0.00353	-0.00172	-0.00000	0.00002
7	2 0 0	0.9826	-12.311	-0.00509	-0.00235	0.00009	-0.00007
71	2 0 0	0.9726	-12.464	-0.00161	-0.00078	-0.00001	-0.00000
72	2 0 0	0.9694	-12.513	-0.00956	-0.00458	-0.00012	-0.00009
8	2 0 0	0.9800	-12.296	-0.00226	-0.00109	-0.00000	-0.00000
81	2 0 0	0.9719	-12.246	-0.00427	-0.00206	-0.00004	-0.00002
82	2 0 0	0.9678	-12.220	-0.00374	-0.00183	-0.00008	-0.00001
83	2 0 0	0.9669	-12.256	-0.00280	-0.00134	-0.00003	-0.00003
84	2 0 0	0.9664	-12.273	-0.00166	-0.00079	-0.00002	-0.00002
85	2 0 0	0.9702	-12.235	-0.00912	-0.00441	-0.00014	-0.00006
9	2 0 0	0.9772	-12.279	-0.00469	-0.00227	-0.00006	-0.00003
10	2 0 0	0.9753	-12.267	-0.00245	-0.00118	-0.00003	-0.00001
73	2 0 0	0.9668	-12.496	-0.00360	-0.00174	-0.00006	-0.00002
101	2 0 0	1.0239	-10.626	-0.00115	-0.00058	0.00000	0.00002
102	2 0 0	1.0162	-10.983	-0.00145	-0.00072	0.00002	0.00002
121	2 0 0	1.0158	-10.979	-0.00243	-0.00119	-0.00001	-0.00001
122	2 0 0	1.0152	-10.977	-0.00060	-0.00029	-0.00000	-0.00000
103	2 0 0	1.0078	-11.282	-0.00280	-0.00137	-0.00000	0.00001
104	2 0 0	1.0042	-11.410	-0.00236	-0.00116	-0.00000	0.00002
105	2 0 0	1.0018	-11.497	-0.00344	-0.00163	0.00003	-0.00002
151	2 0 0	1.0000	-11.503	-0.00259	-0.00125	-0.00002	-0.00001
152	2 0 0	0.9993	-11.517	-0.00489	-0.00236	-0.00003	-0.00002
153	2 0 0	1.0002	-11.502	-0.00358	-0.00173	-0.00002	-0.00001
106	2 0 0	0.9946	-11.518	-0.00478	-0.00232	-0.00005	-0.00001
160	2 0 0	0.9938	-11.602	-0.00004	-0.00002	0.00004	0.00002
161	2 0 0	0.9678	-12.169	-0.00443	-0.00214	-0.00004	-0.00002
162	2 0 0	0.9537	-12.487	-0.00513	-0.00246	-0.00009	-0.00006
107	2 0 0	0.9922	-11.525	-0.00220	-0.00106	-0.00001	-0.00001
108	2 0 0	0.9875	-11.540	-0.00087	-0.00043	-0.00000	-0.00000
109	2 0 0	0.9845	-11.550	-0.00065	-0.00032	-0.00000	-0.00000
110	2 0 0	0.9792	-11.567	-0.00598	-0.00288	-0.00006	-0.00003

Output - First Test