THE UNIVERSITY OF MICHIGAN

SYSTEMS ENGINEERING LABORATORY

Department of Electrical Engineering
College of Fngineering

SEL Technical Report No. 40

OPTIMAL STATIC CAPACITOR ALLOCATION BY DISCRETE PROGRAMMING

by

James B. Young

June 1969

sponsored by

CONSUMERS POWER COMPANY
Jackson, Michigan

This report was also a dissertation
submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy
in
The University of Michigan

ACKNOWLEDGMENTS

I would like to express my gratitude and appreciation to Professors Keki B. Irani, William Root, John Carey, and Franklin Westervelt, for serving on my doctoral committee. In particular, I would like to thank Dr. Irani and Dr. Root for serving as co-chairmen. Their advice and council is greatly appreciated.

I would also like to acknowledge the aid of staff members of the Consumers Power Company. Bill Esser spent considerable time and effort in supplying me with a load flow program that would interface properly with the program developed in my research. His aid was invaluable. In addition, the interest and suggestions of Dave Tarsi, Prateep Ghose, Tony Pretelt, Mike Miller, Walt Bury and Dave Duncan are greatly appreciated. The research itself has been sponsored by the Consumers Power Company through The University of Michigan Systems Engineering Laboratory.

The continuing faith and encouragement of my parents, Mr. and Mrs. I. Carson Young, are deeply appreciated. The efforts represented by this dissertation are dedicated to them. Special appreciation also goes to Libby Matsuhiro for her encouragement in the past four years.

Finally, I would like to thank Sharon Bauerle for the fine work done in typing the manuscript and Don Townsend for the beautiful photographic reductions of the program listings.

TO MY PARENTS

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES (Continued)

ABSTRACT

In electric power transmission low voltages may occur at various points in the transmission network due to large loads or transmission line and generation outages. Under these conditions, shunt static capacitor banks are frequently an economical means of maintaining voltage levels. In this thesis, the problem of determining the location and size of these banks is formulated as a discrete programming problem and an appropriate algorithm and computer program for its solution are presented. Fixed and/or switched banks can be considered and the allocation requiring the minimum capital outlay is determined. The solution process is based upon "implicit enumeration" and is such that suboptimal solutions may be generated in the process of obtaining the optimum. If so desired, suboptimal solutions can even be searched for directly. This is important because not all significant design considerations can be reduced to a dollar value and the best overall design may not be the one which is least costly. In addition, the solutiom method allows more than one of the possible low voltage conditions to be considered simultaneously. This permits efficient allocation of capacitor units so that they assist voltage levels under as many of the anticipated contingencies as possible. An economical design is thereby achieved by lessening the chances of "over-designing" the system. An important attribute of the method is that it is not dependent upon the structure of the model used to determine the changes in system voltages resulting from the capacitor additions. Therefore any future

improvements in system analysis can be utilized with the same program.

Chapter 1

INTRODUCTION

Electric utilities are required to supply all loads in the geographical area covered by their franchise. The exponential increase in the demand for electrical power has therefore forced a tremendous expansion rate upon them. Governmental control of their rate structure has imposed further restrictions so that each system addition must be economical as well as beneficial. And above all, the system design must be such that AC frequency and voltage magnitudes are maintained within fixed tolerances and a high level of continuity of service is guaranteed.

These considerations point out the magnitude of the problems facing the power system designer. He must frequently carry out an extensive examination of many design alternatives before a final solution is chosen. As many alternatives as possible are considered in order that the "best" design from the standpoint of economics, reliability and performance, can be selected. Here the analytical difficulties presented by the large size of modern power networks usually force compromises upon the designer. Typically his studies must be limited and only approximate.

The advent of the modern digital computer has allowed many of the power system analysis problems to be handled automatically. This has resulted in better designs through increased accuracy and has permitted more alternative designs to be considered with a given amount of time and money. In the late 1950's and early 1960's interest started to develop in automating even the selection of design alternatives to be analyzed. This interest was motivated by a desire to find the "best" design alternative(s) and to make more efficient use of the engineer's time by relieving him of burdensome data preparation. This thesis discusses one such "automated design" method.

In electric power transmission low voltages may occur at various points in the transmission network due to large loads or transmission line and generation outages. Under these conditions, shunt static capacitor banks are frequently an economical means of maintaining voltage levels. In this thesis, the problem of determining the location and size of these banks is formulated as a discrete programming problem and an appropriate algorithm and computer program for its solution are presented. Fixed and/or switched banks can be considered and the allocation requiring the minimum capital outlay is determined. The solution process is based upon "implicit enumeration" and is such that suboptimal solutions may be generated in the process of obtaining the optimum. If so desired, suboptimal solutions can even be searched for directly. This is important because not all significant design considerations can be reduced to a dollar value and the best overall design may not be the one which is least costly. In addition, the solution method allows more than one of the possible low voltage conditions to be considered simultaneously. This permits efficient allocation of capacitor units so that they assist voltage levels under as many of the anticipated contingencies as possible. An economical design is thereby achieved by lessening the chances of "overdesigning" the system. An important attribute of the method is that it is not dependent upon the structure of the model used to determine the changes in system voltages resulting from capacitor additions. Therefore any future improvements in system analysis methods can be utilized with the same program. In the following chapters these considerations will be discussed in detail.

It is anticipated that some readers may not be familiar with the practices and terminology of electrical power engineering. In the interest of providing more self-contained reading for them, Chapters 2 and 3 contain general background material. Those familiar with power engineering should skip directly to Chapter 4, which provides a summary of the major contributions to automated capacitor allocation. Chapter 5 begins the discussion of a new approach to capacitor allocation by formulating the problem as one of discrete optimization. It also describes the solution methods available for problems of this type. In Chapter 6 algorithms appropriate for the solution of allocation problems are developed. These algorithms provide the basis for a computer program for optimal capacitor allocation. The problems associated with their computerization are discussed in Chapter 7. An actual program utilizing the algorithms is then discussed in Chapter 8. Details of the program structure, results of its use on test problems and possible extensions and modifications of it are covered in this chapter. Program listings are contained in the Appendix. Finally, Chapter 9 concludes the thesis with a summary of the major contributions and some suggestions for further research in the area of automated capacitor allocation.

Chapter 2

GENERAL BACKGROUND

This chapter briefly describes material that is basic to an understanding of the problems encountered in steady state analysis and design of electrical power systems. Section 2.1 briefly discusses power system structure and general principles of their operation. Section 2.2 introduces a method of power system representation based on "per unit" quantities. Their use greatly simplifies problems of power system analysis. Techniques of network analysis and appropriate matrix notation are reviewed in Section 2.3. A review of AC power relations is given in Section 2.4. Section 2.5 discusses an important analysis tool in power engineering—the "load flow program." This program will be referred to frequently in the following chapters. Then Section 2.6 briefly describes techniques of power system design and some of the work which has been done in automated design procedure. And finally, Section 2.7 discusses the role of automated design procedures.

2.1 Power Network Structure and Operation

The structure of a typical power pool can be broken down into four levels ([40], p. 3).

1) Distribution level
2) Subtransmission level
3) Transmission level (which with its associated subtransmission and distribution networks forms one power system)
4) Tie line system (which connects a number of power systems into a power pool).

Each of these is represented by different voltage levels which are separated from one another by "substations" where transformers change the voltage magnitude. The lowest voltage levels (for example 120V, 4KV, 11KV) belong to the distribution system and their function is to distribute energy to domestic and small industrial or commercial customers. The subtransmission level operates at an intermediate voltage level (typically 46KV or 66KV) and handles larger quantities of power than does the distribution system. It distributes energy to certain industrial customers and also to distribution substations where the voltage is reduced to the distribution level. The highest voltages on the system (138KV, 345KV and up) belong to the transmission level. It not only handles larger quantities of power than the subtransmission level, but it also interconnects all of the generating stations of the system. Normally no interconnection between power plants exists through the subtransmission level.

The classical way of operating a power system is essentially as follows ([40], p. 382). Voltage is maintained at certain levels at specific points in the network. Then all of the generating stations except one are scheduled to provide a specified amount of the load power requirements. The output of the remaining station is varied to take care of fluctuations in the system loads. By doing so, this station maintains the AC frequency of the whole system.

The actual configuration of the power system varies to some extent. Lines or transformers may be removed for maintenance or because of faults within them. In addition, generating units must be shut down occasionally for preventive maintenance. Under such conditions, the burden of supplying adequate power to the consumers falls on the remaining portions of the system.

A power system must be designed for satisfactory operation under both steady state and dynamic conditions. Strictly speaking, the "steady state" of the system never really exists as loads are continually changing. But the term is used to refer to conditions when the system configuration, generation and loads are not undergoing large, rapid changes. Methods of computer analysis have been developed for both the steady state and dynamic performance of the system.

The chapters to follow will be concerned with the steady state design of the power system and so the next few sections will be devoted to steady state analysis. The voltage magnitude, $|E|$, at the system buses[†] will be of primary importance.

2.2 The Per Unit Method

In later sections reference will occasionally be made to the so-called "per-unit" method and "per unit" quantities. Therefore a brief introduction to these concepts is in order. The reader who is interested in looking further into these topics is referred to Chapter 8 in [40] and pages 159-168 in [34].

The solution of voltage and current relations on an electrical power network is complicated by the presence of transformers within the network. As discussed in Section 2.1, these transformers separate the different voltage levels of the system. In principle, one set of equations could be written for the voltage and current relations on each of the voltage levels and a second set could relate the primary and secondary voltage and current of any transformers in the network. The process of solution would, however, be quite involved. The per unit method simplifies the situation by eliminating the need for the set of equations which relate primary and

----

[†] A "bus" is the junction point or "node" for various lines on the system.

secondary quantities of the transformer.

Figure 1a shows an equivalent circuit for a single phase transformer in which the magnetizing current (normally represented by a shunt inductance across the coils) has been neglected. The following equations are seen to apply.

$$E'_p = \frac{n_p}{n_s} E'_s, \qquad I_p = \frac{n_s}{n_p} I_s \tag{2.1}$$

$$E'_s = E_s + I_s Z_s, \qquad E_p = E'_p + I_p Z_p \tag{2.2}$$

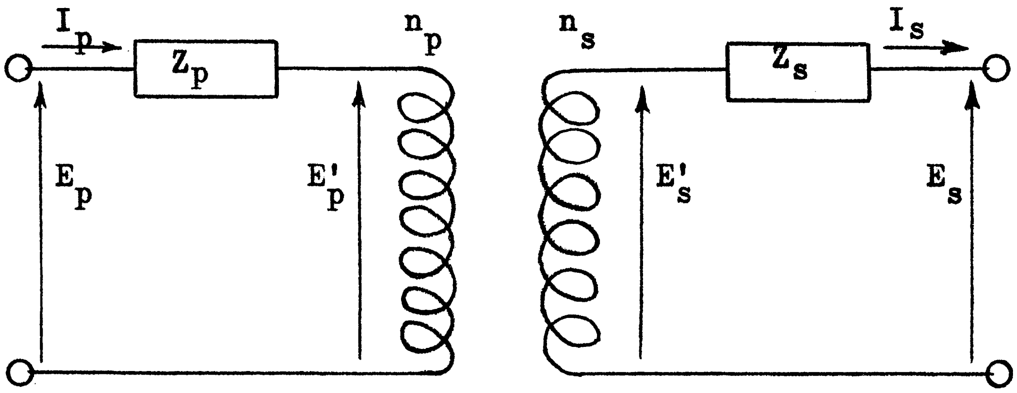
n_p and n_s are the number of turns on the primary and secondary coils, respectively.

Substituting E'_p and E'_s from equations (2.2) into the first equation of (2.1) gives

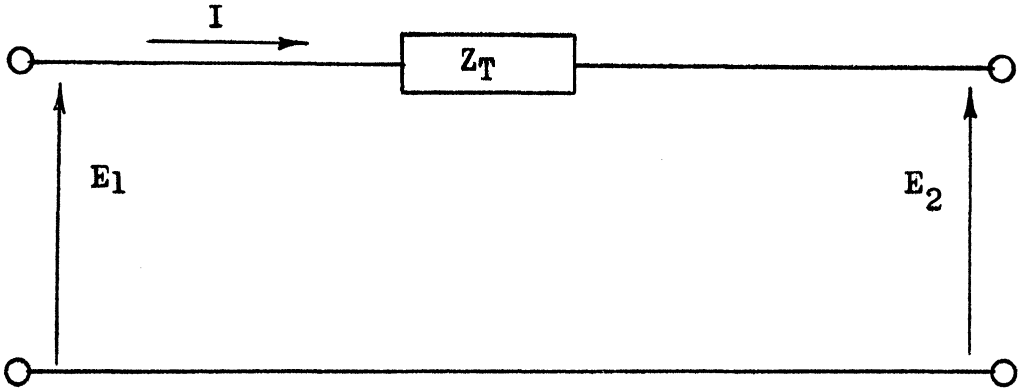
$$E_p - I_p Z_p = \frac{n_p}{n_s} E_s + \frac{n_p}{n_s} I_s Z_s \tag{2.3}$$

This equation can be simplified by expressing each quantity involved as some multiple of a "base" quantity. Let $|E_{bp}|$, $|E_{bs}|$, $|I_{bp}|$ and $|I_{bs}|$ be the base magnitudes for the primary and secondary voltage and currents respectively. These quantities must satisfy the transformer transformation ratio. In other words

$$|E_{bp}| = \frac{n_p}{n_s} |E_{bs}| \tag{2.4}$$



(a) Equivalent circuit of transformer, neglecting magnetizing current.



(b) Equivalent circuit of transformer when quantities are given in per unit.

FIGURE 1

TRANSFORMER EQUIVALENT CIRCUITS

and

$$|I_{bp}| = \frac{n_s}{n_p} |I_{bs}| \tag{2. 5}$$

must hold.

Division of the left and right hand sides of (2. 3) by the left and right hand sides of (2. 4), respectively, gives

$$\frac{E_p}{|E_{bp}|} - \frac{I_p Z_p}{|E_{bp}|} = \frac{E_s}{|E_{bs}|} + \frac{I_s Z_s}{|E_{bs}|} \tag{2. 6}$$

which can be rearranged to give

$$\frac{E_p}{|E_{bp}|} = \frac{E_s}{|E_{bs}|} + \frac{I_p}{|I_{bp}|} \frac{Z_p}{\frac{|E_{bp}|}{|I_{bp}|}} + \frac{I_s}{|I_{bs}|} \frac{Z_s}{\frac{|E_{bs}|}{|I_{bs}|}} \tag{2. 7}$$

Since $I_p = \frac{n_s}{n_p} I_s$ and $|I_{bp}| = \frac{n_s}{n_p} |I_{bs}|$ it is seen that

$$\frac{I_p}{|I_{bp}|} = \frac{I_s}{|I_{bs}|} \tag{2. 8}$$

Use of equation (2. 8) and the noation $|Z_{bp}| = \frac{|E_{bp}|}{|I_{bp}|}$, and $|Z_{bs}| = \frac{|E_{bs}|}{|I_{bs}|}$ in equation (2. 7) gives

$$\frac{E_p}{|E_{bp}|} = \frac{E_s}{|E_{bs}|} + \frac{I_p}{|I_{bp}|} \left(\frac{Z_p}{|Z_{bp}|} + \frac{Z_s}{|Z_{bs}|} \right) \tag{2. 9}$$

At this point it is convenient to introduce the following notation

$$\frac{E_p}{|E_{bp}|} = E_1 \quad \text{per unit primary terminal voltage}$$

$$\frac{E_s}{|E_{bs}|} = E_2 \quad \text{per unit secondary terminal voltage}$$

$$\frac{I_p}{|I_{bp}|} = \frac{I_s}{|I_{bs}|} = I \quad \text{per unit current through transformer} \tag{2.10}$$

$$\frac{Z_p}{|Z_{bp}|} = Z_1 \quad \text{per unit primary impedance}$$

$$\frac{Z_s}{|Z_{bs}|} = Z_2 \quad \text{per unit secondary impedance}$$

$$Z_T = Z_1 + Z_2 \quad \text{per unit transformer impedance}$$

With this notation equation (2.9) becomes

$$E_1 = E_2 + I(Z_1 + Z_2) \tag{2.11}$$

The importance of this conversion process lies in the simplicity of the equivalent circuit for this equation (Figure 1b). The transformer appears as a simple series impedance.

The per unit method can be used to simplify the representation of all the transformers present on a power network. A common base power is chosen for the whole network and a base voltage is chosen for one of the voltage levels present. The remaining base voltages are then determined by equation (2. 4). From the base power and voltage the base values for impedance and current on each voltage level can be determined.

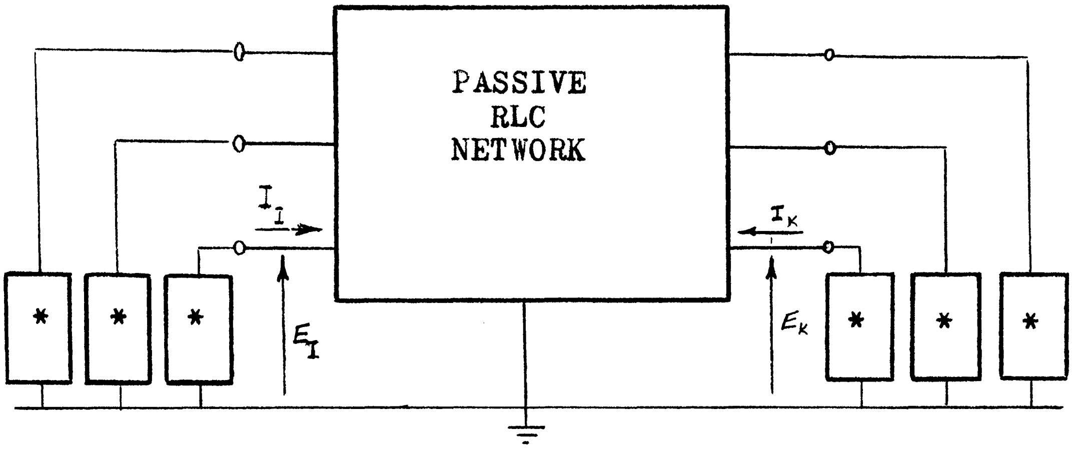
2. 3 Power Network Analysis

As a first step in determining bus voltages on a network with specified generation and loading, a mathematical model of the power network will be developed.

The equations describing voltage and current relationships on an electrical power network can be written with respect to three different reference frames, "bus", "branch" or "loop" ([33], p. 42). The reference frame which is most frequently used in power system studies, and the one which will be considered here, is the bus frame. In the bus frame of reference, the variables are the nodal voltages and the nodal currents of classical network analysis. Consider for the moment, a power network without generators or loads attached. This network will consist of transformers, shunt static capacitors to ground and transmission lines which have series resistance and inductance and shunt capacitance to ground. Thus, using the techniques of the last section it can be modeled as a passive RLC network. Such a network has been represented in Figure 2 as a "black box" in which all nodes or buses have been made available externally. The quantities of interest in the bus frame of reference have been indicated by capital letters. I_k is a complex number representing the magnitude and phase of the AC current injected into bus k of the network as measured relative to the voltage at some (arbitrarily chosen) reference bus. Similarly, E_k is a complex number representing the magnitude and phase of the AC voltage appearing between bus k and ground as measured relative to the voltage of the reference bus. The I's and E's are referred to as bus currents and bus voltages respectively.

Equations relating the bus voltages and bus currents on the system can be obtained through the use of Kirchoff's current law. The network of Figure 3 will be used to demonstrate the method. The symbol $i_{\ell k}$ is a complex number representing the current flowing from bus ℓ to bus k. The admittance of the line between buses ℓ and k is written as $y_{\ell k}$. Kirchoff's current law states that the algebraic sum of the currents flowing into a network node must be zero. Therefore

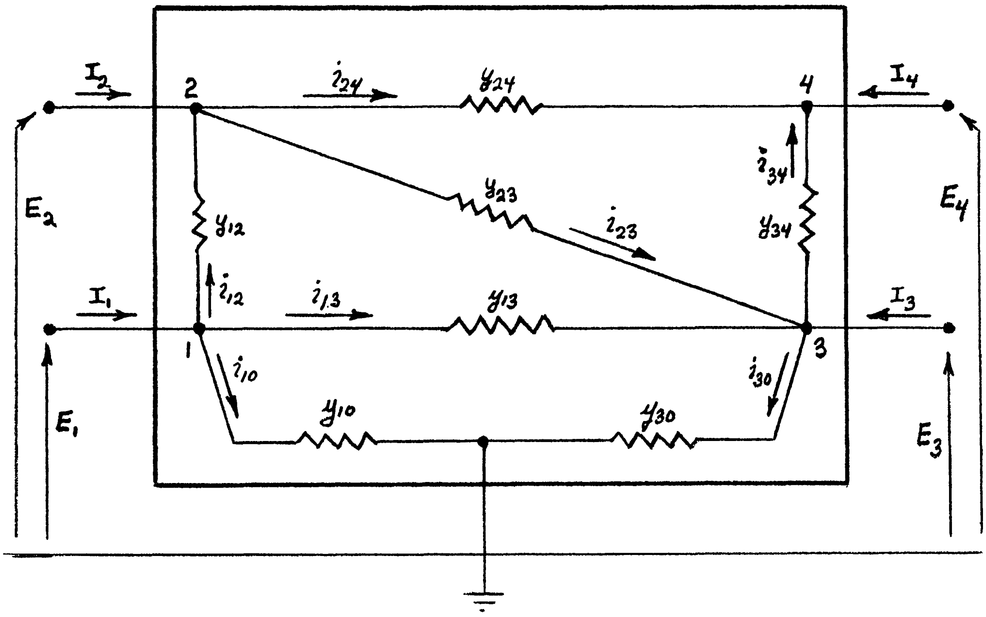
$$\begin{aligned} I_1 - i_{12} - i_{13} - i_{10} &= 0 \\ I_2 + i_{12} - i_{24} - i_{23} &= 0 \\ I_3 + i_{23} - i_{34} - i_{30} &= 0 \\ I_4 + i_{24} + i_{34} &= 0 \end{aligned} \qquad (2.12)$$



* Generation and/or Load

"BLACK BOX" REPRESENTATION OF THE POWER NETWORK

FIGURE 2



EXAMPLE NETWORK

FIGURE 3

Now, each of the line currents can be written in terms of the bus voltages and the line admittances as follows

$$\begin{aligned} i_{12} &= y_{12}(E_1 - E_2) & i_{24} &= y_{24}(E_2 - E_4) \\ i_{13} &= y_{13}(E_1 - E_3) & i_{23} &= y_{23}(E_2 - E_3) \\ i_{10} &= y_{10}E_1 & i_{34} &= y_{34}(E_3 - E_4) \end{aligned} \qquad (2.13)$$

When these are substituted into equations (2. 12) a relation between the E's and I's results

$$\begin{aligned} I_1 &= (y_{10} + y_{12} + y_{13})E_1 - y_{12}E_2 - y_{13}E_3 \\ I_2 &= - y_{12}E_1 + (y_{12} + y_{24} + y_{23})E_2 - y_{23}E_3 - y_{24}E_4 \\ I_3 &= - y_{23}E_2 + (y_{23} + y_{34})E_3 - y_{34}E_4 \\ I_4 &= - y_{24}E_2 - y_{34}E_3 + (y_{24} + y_{34})E_4. \end{aligned} \qquad (2.14)$$

These can be written in matrix form as $\bar{I}_{BUS} = Y_{BUS}\bar{E}_{BUS}$. Each diagonal term of the matrix Y_{BUS} is the sum of all the admittances terminating on the corresponding node and each off-diagonal term is the negative of the line admittance between the two corresponding nodes. Since power networks rarely have more than three or four lines connected to any one bus, this means that for a large power system Y_{BUS} is typically "sparse" in that it contains mostly zero elements off of the diagonal.

In order to obtain an equation giving the bus voltages as a function of the injected bus currents, the <u>bus admittance matrix,</u> Y_{BUS}, can be inverted to give

$$\bar{E}_{BUS} = Y^{-1}_{BUS}\bar{I}_{BUS} = Z_{BUS}\bar{I}_{BUS}.$$

Z_{BUS} is called the bus impedance matrix of the network. As the sparsity of Y_{BUS} is lost in the inversion process Z_{BUS} is usually a "full" matrix with few nonzero elements. The reader should note that the most efficient means of obtaining Z_{BUS} on a digital computer is not by inverting Y_{BUS}. An algorithm now exists with which Z_{BUS} can be constructed directly from the line impedance data ([33], Chapter 4). This algorithm is somewhat involved and will not be discussed here.

Since power transmission equipment presents a largely inductive impedance to the flow of current, the elements of Z_{BUS} are dominantly reactive. In some analysis it is sufficient to consider only the imaginary components of these elements. They are usually written in matrix form as the "bus reactance matrix", X_{BUS}. This approximation will appear in a later chapter.

2.4 AC Power Relations

It will be shown in the next section that power flows on the network must be taken into consideration in order to determine the bus voltages. It is therefore advisable to consider for the moment the types of power that will be encountered.

Consider once again Figure 2. Let the instantaneous bus voltage and bus current at bus K be given by $e(t) = \sqrt{2}\ |E_K| \cos \omega t$ and $i(t) = \sqrt{2}\ |I_K| \cos (\omega t + \phi)$. Then the instantaneous power fed into bus K from the external source is given by

$$p(t) = 2|E_K||I_K| \cos \omega t \cos (\omega t + \phi)$$

or

$$p(t) = (1 + \cos 2\omega t)|E_K||I_K| \cos \phi - |E_K||I_K| \sin \phi \sin 2\omega t$$

The first term in this expression represents power which flows only in one direction. Its time average is $P_K = |E_K||I_K| \cos \phi$ and is referred to as real or active power. The time average of the second term is zero. It represents energy which oscillates between the inductances and capacitances of the system. The quantity $Q_K = + |E_K||I_K| \sin \phi$ is called reactive power and two types are distinguished: inductive reactive power, with the current lagging the voltage in phase (a negative quantity) and capacitive reactive power, with current leading the voltage (a positive quantity). The complex or phasor power into bus K is defined as $S_K = P_K + jQ_K$. It can be easily obtained from the equation $E_K I_K^* = P_K + jQ_K$ in which I_K^* is the complex conjugate of I_K.

2.5 Load Flow Analysis

To analyze the steady state performance of a power system with specified loading and generation, an analysis procedure referred to as a load study is performed. The information obtained from such a study is the magnitude and phase angle of the voltage at each bus. From this information the current or power flows through any of the system apparatus can be calculated. This enables the power engineer to determine whether the existing or anticipated loads can be supplied with adequate voltage and without overloading system equipment. In this section load flow analysis as performed on digital computers is briefly described.[†]

The equation $\bar{E}_{BUS} = Z_{BUS}\bar{I}_{BUS}$ or $\bar{I}_{BUS} = Y_{BUS}\bar{E}_{BUS}$ describes the relationship between current and voltage on the power system network. These equations were derived in such a manner that the generators on the system should be represented either as constant voltage sources or as constant current sources[††]. In the former, the magnitude and phase angle of the generator terminal voltages would be fixed and in the latter the magnitude and phase of the injected current would be. Similarly, the loads should be modeled either as constant current sources or as fixed shunt impedances. Unfortunately, not all of the parameters necessary for these descriptions are available. For example, the generator terminals are frequently maintained at a fixed voltage magnitude, but there is no means of specifying their phase angles. These are unknown and are part of the information desired from the load flow solution. Specification of current sinks for load models experiences similar complications—the phase angles of the currents not being known until the problem has been solved.

[†] A more complete description of this and other methods is given in [33], Chapter 8.

[††] The derivation of Y_{BUS} assumed current sources to be supplying the bus currents.

In practice, the generator is modeled by specifying the voltage magnitude at its terminals and the real power, P, being generated by it. This corresponds to the way generators are actually operated.

The problem of modeling system loads is not an easy one as many different types of equipment are connected to the system. At present, system loads are commonly modeled by their real and reactive power demands. This choice is apparently influenced by the behavior of motor loads as these increase their full load current as their terminal voltage drops.

Thus the problem of solving for bus voltages on the system is one of solving the equation $\bar{E}_{BUS} = Z_{BUS}\bar{I}_{BUS}$ or $I_{BUS} = Y_{BUS}\bar{E}_{BUS}$ subject to certain nonlinear constraints on E and I at each bus. These constraints are of the form $EI^* = P + jQ$ at the load buses while at the generator buses the parameters $|E|$ and P are held fixed. It is also necessary to select one bus, called the slack bus, to provide the additional real and reactive power needed to supply transmission losses, as these are not known until a solution for the bus voltages has been obtained. This bus is chosen as the reference bus and complex E is specified for it.

At present, the solution of these equations is found by iterative techniques performed on a digital computer—there being no other convenient way of solving the nonlinear algebraic equations.

All bus voltages are set to approximations of their anticipated voltages and then successive corrections to these approximations are computed. When the corrections become less than some specified tolerance, the existing approximation is considered to be the solution. There are many techniques available for calculating the voltage corrections and each has its advantages and disadvantages (See [32], Chapter 8). Unfortunately, in some cases, the corrections may increase in magnitude indefinitely and the iterations will not converge to a solution。 Such instability can sometimes be prevented and the speed of convergence to a solution improved by providing a better initial guess for the bus voltages. (Usually these are all set equal to the slack bus voltage.)

Present day load flow analysis is almost completely automated. The engineer prepares data on lines and transformers, etc. and submits this to the computer along with the configurations, generation and loading he wishes analyzed. The computer program automatically forms Z_{BUS} or Y_{BUS} and then iteratively solves for the bus voltages. Then these voltages and data on losses and power current flows are listed on an output printer. Use of the Y_{BUS} formulation is quite common in these programs because the sparsity of Y_{BUS} makes for economical use of computer memory.

2.6 Power System Design

In the introduction in Chapter 1 it was mentioned that the government places economic constraints on the utility industry in the form of rate controls. In addition, the financing of major system modifications is not always easily accomplished and this results in further economic constraints. In the face of continual increases in consumer power demands, the utilities must provide an economical plan of expansion for their systems. This means that any network design selected must be economical at the time it is realized and also economical in view of its effects on future design policies. It is interesting to note that a most economical long term design may be impractical due to excessively high initial costs. Hence, the long term design problem must necessarily be based on economical short term designs. The most commonly used approach to these problems is to obtain a set of feasible solutions for each of a series of the time periods extending into the future. Then the best solution is determined for each period by considering the sequence as a whole.

At present, this procedure of "policy comparison" is accomplished by a sizable amount of data manipulation on the part of the system planning engineer. Relying upon his knowledge of the system, he will select and analyze a set of design possibilities for each of the design periods in question. This involves, in addition to the preparation of system data, a considerable amount of interaction with the computer analysis programs. This is continued until satisfactory design candidates are obtained. In some cases, even after obtaining satisfactory candidates, the planner cannot be sure that he has found the "best" candidates available. Time and expense limit the extent of his search. For this reason, interest has developed in automating the search for the "best" design possibilities. This approach holds the promise of providing better designs through more thorough search procedures—and with a savings in the time required to obtain them.

The methods used to date for automating the short and long term aspects of design might best be broken down into two categories: those based on mathematical techniques and those based on heuristic techniques. The heuristic methods vary from programmed manual procedures and "rules of thumb" [1, 4, 7, 14, 23, 26, 29, 37, 38] to sophisticated adaptive methods which modify their approach to a given design problem in accordance with computational experience [12]. Heuristics are only claimed to aid in finding some of the "better" design possibilities.

Variational calculus has been applied to some special design problems such as the allocation of capacitor units to minimize losses in transmission lines [8, 32]. However, calculus is not appropriate for many design problems as such things as the configuration of the networks are not continuously variable.

Gaussens and Calvert [16] have phrased the problems of economic expansion and operation of power networks as a mathematical programming problem. An optimal solution is sought in the sense of one that minimizes an established cost function. They point out that no optimal sequence of equipment additions can be determined except for simplified cases in which the possibilities are limited. They propose two methods of solution for the simplified problems they consider. One is based on manually directed search, the other is automated search based on heuristics.

Some design problems can be formulated as mathematical programming problems for which rigorous solution methods are known. This usually requires that some simplifying assumptions be made. Knight [19] describes the use of linear programming in determining minimum cost policies for network expansion, while Oldfield [28] presents a dynamic programming approach to economic long term planning once sets of suitable designs have been determined for each interval in time. Dynamic programming has also been applied to capacitor allocation for the purpose of minimizing losses on radial lines [10]. These methods succeed in finding an optimum solution based on a mathematical model of the system. The actual optimality depends upon the validity of the mathematical model.

2. 7 The Role of Automated Design

It is this author's belief that the role of automated design techniques, such as the above, is one of supplementing the human designer and not one of replacing him. Many of the trade-offs which must be taken into account would be very difficult to include in a mathematical model and only human judgment can be relied upon to make the final decisions. But automated design techniques can aid the designer in several ways:

1) They can reduce the amount of time he spends in data preparation.

2) They can reduce the time required to obtain a satisfactory solution.

3) In many cases they can provide him with the least costly or optimal solution from the standpoint of some established cost function.

4) They can provide him with a list containing the "best" solutions from the standpoint of the cost function. This list can then be searched for the appropriate solution in view of the designer's total knowledge of the problem.

It should be noted that there are many considerations in power system design which cannot easily be taken into account on a dollars and cents basis. The importance of an equipment installation being aesthetically pleasing to the community is a good example. System reliability is another. This makes some means of obtaining suboptimal solutions an important asset of any automated design procedure. If the optimum solution is not entirely satisfactory, a selection can be made from the set of suboptimal solutions listed. The knowledge of the optimum solution from the standpoint of the cost function can be used to determine how much one is sacrificing by allowing things like aesthetics and reliability to affect one's decision.

Chapter 3

SUMMARY OF THE VOLTAGE REGULATION PROBLEM

In this chapter some of the problems of steady state design are discussed with particular emphasis on the problem of voltage maintenance and the use of shunt static capacitors for its solution. Section 3. 1 provides an introduction to voltage requirements and in Section 3. 2 methods of meeting these requirements are discussed. Section 3. 3 describes some of the beneficial effects of shunt capacitors. This is followed in Section 3. 4 with a discussion of possible undesirable effects accompanying shunt capacitor use. Section 3. 5 concludes the chapter by discussing the role of shunt capacitors in voltage control and by summarizing the capacitor allocation problem.

3. 1 Voltage Requirements

Due to the changing character of the system loads the power engineer must continually update all levels of the power system in order to assure satisfactory performance. The loads vary to some extent from instant to instant and may change considerably over the course of years. The system is always designed to handle the short-term load variations, but its design must be continually reevaluated in view of the long-term changes. The load limits of an existing network are frequently determined by one of two factors: the minimum allowable delivered voltage or the current carrying capacity of the equipment involved.

Almost all present day electrical equipment is designed to operate at a certain terminal voltage. Operation at voltage other than that specified will generally result in diminished performance. For example, reduced voltage at motor terminals can reduce starting torque and increase the full load current. On the other hand, excessively high voltage may shorten equipment life time. Since it is not economically possible to maintain an absolutely constant voltage at every customer's service terminals, a band of permissible voltages is usually established by the Public Service Commissions of the various states. The problem of maintaining voltages within this band involves every level of the integrated power system. The selection and coordination of proper equipment for maintaining and regulating voltage is frequently one of the major problems of the power engineer. The problem is complicated by the large range of system conditions over which voltage must be maintained. Loads may undergo large daily fluctuations in some areas. In addition, the removal of certain "critical" pieces of equipment must be anticipated. These removals or "outages" could include the removal of lines or transformers during fault conditions or the removal of generating units for maintenance. Overcompensation of voltage under such conditions may result in excessively high voltages when more normal conditions prevail—unless, of course, proper design measures are taken.

As might be expected, load flow analysis plays an important role in designing a system to meet these conditions. Typically, each possible design is analyzed under each of a finite number of system states or conditions. A design must give satisfactory performance in each case to be acceptable. The selection of the system conditions to be considered is not an easy task. At present, the most "critical" conditions are selected on the basis of the planning engineer's experience.

3.2 Voltage Control Methods

The problem to be considered in later chapters is the maintenance of the system bus voltages within certain specified ranges under a finite set of system conditions which are considered to represent the extremes in voltage levels anticipated. In general, there are three methods of accomplishing this: ([40], p. 462).

1) Addition of sources of reactive power such as synchronous condensers and shunt static capacitors.
2) Addition of new lines and series capacitors. These reduce circuit impedance.
3) Addition of variable-tap transformers such as "boosters," induction regulators and tap-changing-under-load (TCUL) transformers.

Induction regulators and TCUL transformers are basically transformers which are capable of automatically varying their turns ratios under loaded conditions. Most all transformers used in power networks have provisions for varying their turns-ratio, but some must be de-energized for the operation to be performed. The automatic transformers provide a voltage regulating ability while the non-automatic types are limited to voltage-boost applications. In either case, the range of input voltage over which a relatively constant output voltage can be maintained, is limited. Frequently, this range is in the order of ± 10% of the nominal voltage of the unit.

Line additions present two major problems. First, appropriate line additions must be determined with load flow studies—a task not always easily performed. Then the problem of securing appropriate right-of-ways for the lines presents itself. This is at times hindered by appearance conscious citizens who object to "unsightly" poles and transmission towers.

Series capacitors are presently in use on a number of power systems, but their use is not as yet wide-spread. In some respects they are still in the experimental stage.

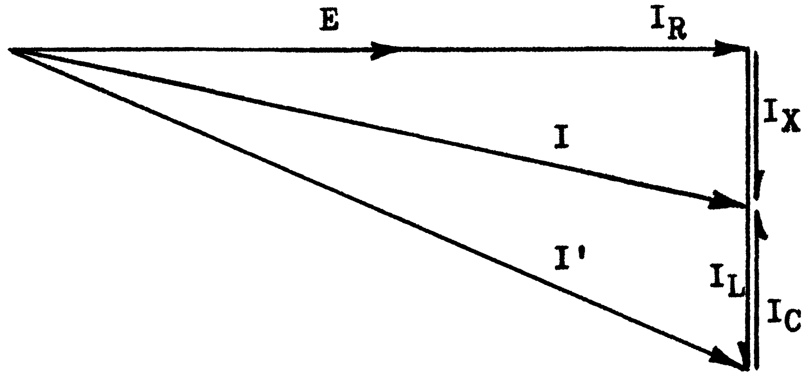
Synchronous condensers are basically normal synchronous machines with no prime mover or mechanical load attached to them. They behave as a source of reactive power when their excitation voltage is sufficiently large. In this respect, they behave much as a shunt static capacitor except that their reactive power output is continuously variable. They have the disadvantage of being more expensive and of having larger losses than static capacitors. The reader is referred to Chapter 9 of Zaborszki [40] for a more complete evaluation of the advantages and disadvantages of synchronous condensers.

It is common policy for utilities to try to delay large capital outlays for equipment as long as possible. For this reason, the development of relatively inexpensive static capacitor units has made them a favored choice in attempts at solving several types of system problems—including those of voltage maintenance and regulation. These static capacitor units, utilized in shunt connections to ground, are the major concern of this thesis. Some of the benefits which result from their use are discussed in the next section.

3.3 Beneficial Effects of Shunt Capacitors

Most of the equipment in an electrical power system presents a largely inductive impedance to the flow of AC current. In addition, most of the loads draw a large amount of inductive (or magnetizing) current. This means that there are several benefits that can result from the use of shunt static capacitors. They are all a direct consequence of the property that the current through a capacitor applied in shunt with a load is 180^o out of phase with the current through the inductive component of the load. Figure 4 shows how the total reactive current is drawn by the capacitor-load combination is less than that drawn by the load alone as a result of this cancellation.

One important ramification of the reduction in total reactive current drawn is that the magnitude of the total load current is also reduced. (See Figure 4.) This has two desirable effects on the



I_R current drawn by real component of the load.

I_L current drawn by inductive component of the load.

I_C current drawn by capacitive component of the load.

I_X total reactive current drawn with capacitive component present. Without capacitive component I_X I_L.

I' net current drawn by real and inductive components of the load.

I net current drawn by real, inductive and capacitive components of the load.

REACTIVE CURRENT CANCELATION

FIGURE 4

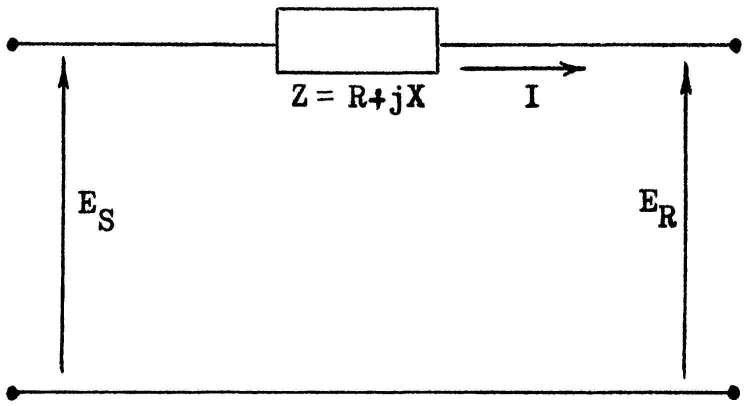
system. First, there is a reduction in both real and reactive power losses ($P + jQ = R|I|^2 + jX|I|^2$) in all equipment through which this current is drawn. The loss reductions in real power can be related directly to fuel savings. Second, when current is the limiting factor which determines the maximum load supplied through equipment, larger loads can be supplied through existing equipment if the current drawn by them is reduced. The addition of shunt capacitors can thus delay large capital expenditures for new lines and transformers.

In some instances delivered voltage is the factor which limits the amount of load that can be supplied by existing equipment. Again, capacitor additions can sometimes delay line and transformer additions by raising the delivered voltage. When current containing an inductive (90^o lagging) component is drawn through lines and transformers possessing inductive impedance the interaction of the inductive current and the inductive impedance results in a voltage drop. This is shown in Figure 5b for a short transmission line. The equivalent circuit of the short transmission line is shown in Figure 5a . When an appropriate number of capacitors are placed in shunt with the load at the receiving end of the line, the magnitude of the reactive current drawn can be reduced. This can in turn reduce the voltage drop along the line as shown in Figure 5c.

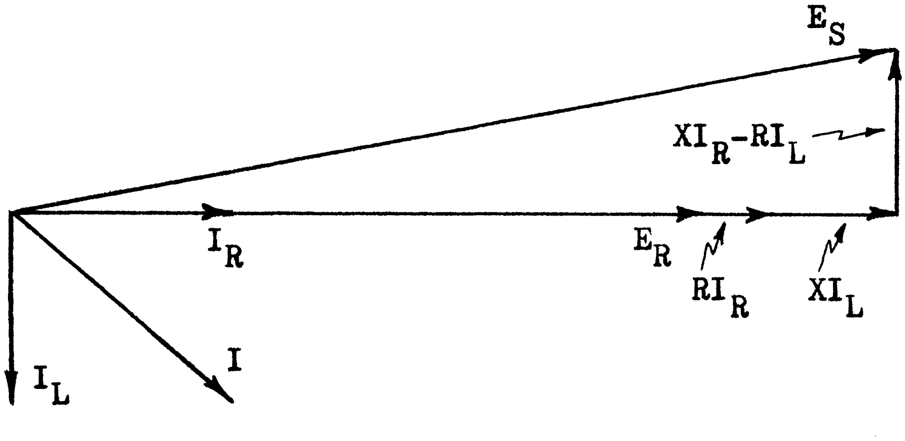
It should be noted that when inductive current flow is corrected near the loads which are drawing large amounts of it, the benefits which result are felt on all parts of the power system which supply

PHASOR DIAGRAMS FOR VOLTAGE RELATIONS ON A SHORT TRANSMISSION LINE

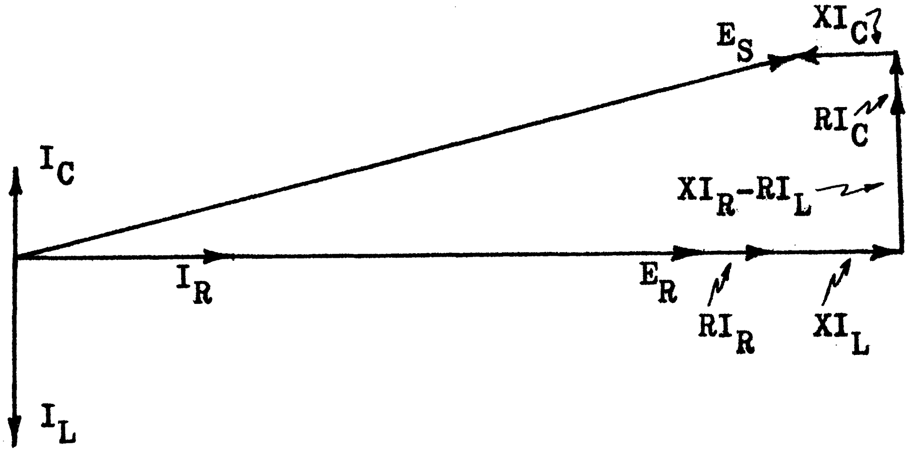
FIGURE 5



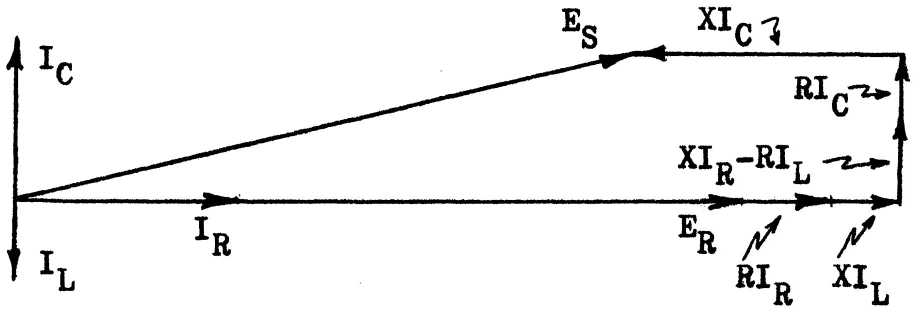
(a) Equivalent circuit of the short transmission line.



(b) Lagging power factor load.



(c) Lagging power factor load with capacitor correction.



(d) Lagging power factor load with capacitor correction. Note that $|E_R|>|E_S|$.

power to these loads. This means that if the loads on the distribution system are corrected with shunt capacitors, the beneficial effects of this correction will extend into both the subtransmission and transmission levels. An important advantage of static capacitors is that they can be economically applied even in small units and in a decentralized fashion at as low a level of the system as the distribution level. Synchronous condensers are far too expensive and cumbersome to be applied in this manner.

The product of reactive current magnitude and the line voltage magnitude is expressed in units of volt-amperes reactive or VARS. This is the same unit as that used to measure reactive power, Q. Since the current through a capacitor is 180^o out of phase with that through an inductor, power engineers usually speak of a shunt capacitor as "supplying VARS or reactive power to the system." The net power factor of a capacitor in parallel with a load is determined by the extent that the load's lagging volt-amperes are supplied by the capacitor. Power capacitors are therefore rated in terms of the kilovars they supply at some rated voltage and the capacitor units are manufactured in certain KVAR sizes. If larger sizes are required, units are connected in parallel to form "banks." Unit sizes which might be encountered on a 46KV subtransmission on system are 7. 2, 8. 4 and 9. 6 MVAR. These can cost upwards of $30, 000 a piece.

3.4 Possible Undesirable Effects of Shunt Capacitors

It is possible for too much capacitance to be placed in shunt with the load. If the net load should become sufficiently capacitive, a situation may arise in which $|E_R| > |E_S|$. (See Figure 5d.) This can happen under light load conditions when capacitors are permanently connected to the system or even on long high voltage lines which naturally poses large shunt capacitance to ground. This effect sometimes necessitates that only switched or disconnectable units be used.

Actually, the magnitude of system voltage is only one aspect of its adequacy. The magnitude and speed of any variations in system voltage are also important. When capacitor banks are switched in and out of a network, they can produce very undesirable fluctuations in voltage. It is therefore necessary to set a maximum size for each switched capacitor bank in order to limit the size of these variations. A limit in voltage change of 4.5% of the nominal voltage is sometimes set. Even in the application of fixed capacitor units it may be desirable to limit the voltage rise of a bank for reliability purposes.

3.5 The Role of Shunt Static Capacitors in Voltage Control

Variations in system operating conditions are large enough that voltage regulation cannot be accomplished by equipment installed at only one of the voltage levels. In addition, no single type of equipment is a panacea for the system's voltage ills. Each has its own particular limitations. For example, voltage regulating transformers can regulate their output only over a limited range of input voltage. And they do require inductive current for their own operation, thereby aggravating the voltage drop prior to their input terminals. On the other hand, shunt static capacitors correct for inductive current demands, but they can also cause excessively high voltage in some instances and they cannot continuously vary their correction. The shortcomings of some other types of equipment are primarily economic. This is the case with line additions and synchronous condensers. In short, the problem of maintaining customer supply voltage within a permissible range involves the proper coordination of all types of voltage control equipment at all levels of the power system.

Shunt static capacitors are frequently an economic means of correcting voltage sag at portions of the power network which are removed from the power supply points. They are used to supplement other types of voltage control equipment, by placing the input voltage to this equipment within a range for which adequate output voltage can be maintained. The voltage profile of a network may undergo large changes due to variations in network configuration and power supply and demand. Such changes may make it necessary for one or more of the installed capacitor banks to be automatically inserted into the network under some conditions and removed from the network under others. The problem of capacitor allocation is one of determining the proper location, size and type (i. e. fixed or switched) of static capacitor banks so that voltage profiles will be maintained within specified limits under all anticipated network conditions. It may be that no arrangement of capacitors alone will meet this need and other types of equipment, such as lines, must be added.

Chapter 4

METHODS OF CAPACITOR ALLOCATION FOR VOLTAGE CONTROL

This chapter summarizes the work done to date on the problem of determining the best locations, sizes and type (either fixed or switched) of capacitor banks to maintain bus voltages within a given range. Section 4.1 discusses some heuristic methods of solution. Then Section 4.2 discusses a constrained optimization formulation of the problem. Section 4.3 concludes the chapter with a brief introduction to the method derived by the author.

4.1 Heuristic Methods

At present, the means of obtaining capacitor allocations that will correct system voltages is largely based on heuristic design strategies. These may either be executed by hand calculation or by computer. Heuristic techniques are a necessity in hand calculations as few electrical systems are simple enough to be rigorously analyzed by hand. Surprisingly enough, even with the computer's computational abilities, one may still be forced to use heuristic approaches when using it for problem solving. This is sometimes the case in design problems where there are a large number of possible approaches to the problem. Even at the computer's high computational speeds it may take too long to consider every possibility.

Thus heuristic decision rules provide a means of reducing the necessary computation.

One heuristic strategy to find a capacitor allocation for voltage control is described by Baum and Frederick [4]. It is apparently designed for manual application, although it could be easily programmed. The circuits it deals with are distribution feeders. These are essentially radial lines tapped for loads at various points. Baum and Frederick's approach is to determine fixed capacitor requirements on the basis of maintaining a nearly flat voltage profile under light load conditions. Capacitors are applied at the lowest point on the voltage profile until restrictions which have been placed on allowable power factor or voltage rise are violated. Then the lowest point on the new profile is considered and the process is repeated. Once the light load voltage requirements have been satisfied in this manner the heavy load conditions are considered. Any additional capacitors required to maintain a nearly flat voltage profile under these conditions are added as switched units that can be automatically removed from the network for light load conditions.

Implicit in most capacitor allocation schemes is the desire to accomplish what correction is necessary at the minimum cost. A rigorous approach should consider even the losses in the lines and their possible reduction with judicious placement of the capacitor units. Unfortunately, this consideration complicates the problem a great deal and, as a result, has not been included in the more complicated capacitor voltage control problems solved to date.

As the network becomes more complicated, so too does the problem of determining the least expensive locations and sizes of capacitor units to maintain voltage. In fact, one may have difficulty in determining whether or not voltage maintenance with capacitors is or is not possible. Instead of just a few possible capacitor allocations, there are many and one's intuitive "feel" for the situation no longer suffices. Some work has been done at the Consumers Power Company on a capacitor allocation computer program which can handle the complexities of larger, more involved networks in an efficient manner. This work has been done independently from the author's although his work was also sponsored by Consumers Power. The program is intended for use at the subtransmission level and considers the application of switched capacitor banks. A form of sensitivity analysis is used whereby the effectiveness of each bus in solving the low voltage problem is determined. Capacitors are first added at the bus having the largest effectiveness rating. Additions are continued there only until the voltage rise resulting from the total bank exceeds a specified maximum. Then the same process is repeated at the bus with the next highest rating. This is continued until all of the low voltage constraints are satisfied.

The bus effectiveness rating is obtained by using the bus reactance matrix, X_{BUS}. The equation $\bar{E} \cong X_{BUS}\bar{I}$ relates bus voltages to the load currents injected at the buses. A capacitor addition can be considered as the injection of an increment of current at the bus to which it is applied. This results in a change in some of the system bus voltages. The incremental change in bus voltages is given approximately by the equation $\overline{\Delta E} \cong X_{BUS}\overline{\Delta I}$, where $\overline{\Delta I}$ is a vector containing all zero elements except for the one corresponding to the bus where the capacitor is installed. The current injected by a capacitor into a bus node, K, is approximately equal to $- jB_c E^o_K$, where B_c is the capacitor susceptance and E^o_K is the (complex) voltage existing at the bus before the capacitor addition.[†] If $(\Delta E_J)_K$ is the approximate voltage rise at bus J due to the capacitor at bus K, an approximate rating of the effectiveness of bus K in solving the low voltage problem can be ovtained by averaging the voltage rises $(\Delta E_J)_K$ over all the buses with low voltage. For example

----

[†] The capacitor current would actually equal $- jB_c E_K$ where E_k is the voltage after the capacitor addition. Since the voltage rise due to a capacitor unit is normally in the order of a few percent of E^o_K, the error resulting from assuming $\Delta I_K \simeq - jB_c E^o_K$ will be in the order of a few percent.

$$\text{Effectiveness of bus K} = \sum_{\forall J \in L} \beta_J (\Delta E_J)_K$$

where L is the set of low voltage buses and β_J is a weighting factor.

Heuristically, this approach appears to be a good one. We would expect it to find one of the least expensive solutions to a particular capacitor allocation problem. However a problem arises in deciding on appropriate weighting factors, β_J. It seems logical to use some sort of weighted average in which the voltage rises at the lowest voltage buses are weighted most heavily. The particular weighting used will certainly determine the cost of the solutions obtained. If one is interested in obtaining the cheapest possible solution, it is not clear which weighting scheme is appropriate.

The capacitor allocation problem is further complicated by the consideration of possible outages of lines, transformers or generators. Once a solution for each outage case has been obtained, they could be combined to form a single solution by installing the largest bank of capacitors found to occur at a given bus in any of the solutions. But this procedure may produce an allocation in excess of actual needs. In some instances smaller, judiciously placed banks may maintain adequate voltage under all of the anticipated outages by sharing VAR capacity through the transmission network. In order to recognize such possibilities, it is necessary to consider all of the expected outage conditions at one time. The effectiveness rating discussed above could be extended for this problem by averaging over the low voltage buses in each of the outage cases. But if one is interested in obtaining the least costly solution, the problem of determining appropriate weighting factors still exists.

4.2 Linear Programming Method

Coincident with the time of this writing, an article has been published which describes a linear programming method for planning kilovar requirements on high-voltage transmission networks [24]†. The problem formulation is such that multiple outage conditions can be considered simultaneously. In addition, an objective function is minimized to obtain the "optimum" solution—in this case the least total VAR generation required to maintain voltages. The method is based on the following linear approximation for the change in bus voltage as a function of bus VAR generation.

$$\Delta|E_I| \cong \sum_J x_{IJ}\Delta Q_J \qquad (4.1)$$

where $\Delta|E_I|$ = per unit change in bus voltage magnitude at bus I

x_{IJ} = imaginary part of the Z_{BUS} (or X_{BUS}) element, Z_{IJ} (or X_{IJ}).

ΔQ_J = change in kilovar generation at bus J

----

†Distribution delayed until March 1969 due to a printer's strike.

Since no mention was made in the article of how equation (4.1) is derived, a few words on its origin may be appropriate at this point.

Consider the equation $\overline{\Delta E} \cong X_{BUS}\overline{\Delta I}$ which as been discussed in Section 2.3. Assume that all parameters and variables are given in per unit values. Remember that the elements of X_{BUS} are all imaginary quantities. Now, if the VAR generation due to capacitors at bus K is known, the current injected into the bus by the capacitors is $-jQ_K/E^*_K$. If Q_K is changed by an amount ΔQ_K, the voltage at the bus will also change by some amount ΔE_K. The change in bus current, ΔI_K, accompanying the change ΔQ_K is given by

$$\Delta I_K = \frac{-j(Q_K + \Delta Q_K)}{(E_K + \Delta E_K)^*} + \frac{jQ_K}{E^*_K} . \qquad (4.2)$$

ΔE_K is not known, but an approximation for ΔI_K can be written as

$$\Delta I_K = \frac{-j\Delta Q_K}{E^*_K} . \qquad (4.3)$$

Using the equation $\overline{\Delta E} \cong X_{BUS}\overline{\Delta I}$, ΔE_I can be written approximately as

$$\Delta E_I \cong \sum_K \frac{(j\, x_{IK})(-\, j\Delta Q_K)}{E_K^*} \tag{4. 4}$$

where $X_{IJ} = jx_{IK}$. This can be further written as

$$|\Delta E_I| e^{j\theta_{\Delta I}} \cong \sum_K x_{IK} \frac{\Delta Q_K}{|E_K|} e^{j\theta_K} . \tag{4. 5}$$

Now, if $\theta_{\Delta I} \cong \theta_I$, then $\Delta|E_I| \cong |\Delta E_I|$. And, if $\theta_K \cong \theta_I$ for all K for which x_{IK} is significantly different from zero, then (4. 5) can be further approximated by

$$\Delta|E_I| \cong \sum_K x_{IK} \frac{\Delta Q_K}{|E_K|} . \tag{4. 6}$$

These assumptions on phase angle are not unreasonable for most buses in the neighborhood of one another will differ in phase less than 10^o. If it is further assumed that $|E_K|$ is not appreciably different from 1 per unit, then

$$\Delta|E_I| \cong \sum_K x_{IK}\Delta Q_K . \tag{4. 7}$$

This is identical to (4. 1).

This equation is used to formulate the VAR allocation problem as a problem in linear programming. $\Delta|E_I|$ becomes the voltage drop which must be corrected at bus I during a particular contingency. The ΔQ_J become the VAR generation required to correct the voltage drops. One equation such as (4. 1) represents each bus for each contingency being considered. The objective of the problem is to minimize the amount of VAR generation needed to correct all the bus voltage drops in all of the contingencies. This can be stated as[†]

Minimize $\sum_J \Delta Q_J$

subject to $\Delta E_I^n = \sum_J x_{IJ}^n \Delta Q_J$ and $0 < \Delta Q_J$

where I ranges over all buses of concern

J ranges over all buses where VAR additions can be made

n ranges over all outage cases considered.

This problem can be solved with any one of several linear programming programs now available.

Once the ΔQ_J required for a given problem have been determined, they can be used to find the static capacitor or synchronous condenser sizes required at each bus. Since both are manufactured only in discrete sizes or ratings, it may be necessary to adjust any fractional unit sizes obtained to the next whole unit. It should be noted, however, that in some cases this rounding off may produce allocations which are not the least expensive. As a

----

[†] Some additional constraints were included in the formulation, but the ones given here will suffice for the purpose of the present discussion.

hypothetical example, consider two adjacent buses at which an optimization based on continuous variables has determined that 1.2 and 1.4 units of capacitance are optimal. Adjusting these values to the next whole unit results in a total of 4 units—2 at each bus. It is conceivable that 2 units at one of the buses and 1 unit at the other might satisfy the voltage constraints, and with a savings of one capacitor unit. The only way that this possibility would be discovered would be to take the discreteness of the condenser sizes into account in the optimization process.

4.3 Integer Programming Method

In the following chapters an integer programming formulation of the capacitor allocation problem is presented. A computer program is developed for determining the least costly allocation of static capacitors that will maintain voltage under specified outage conditions. The method used has the following advantages.

1) It provides a systematic approach to three different types of capacitor allocation problems—those involving only fixed banks, those involving only switched banks, and those involving both fixed and switched banks.

2) It is based on the optimization of a cost functional which is related directly to the cost of capacitor installations in dollars.

3) The discreteness of the variables involved is taken into account in the optimization procedure.

4) No theoretical limit exists for the number of outage conditions that can be considered simultaneously. (The practical limit is set by computer storage available.)

5) Suboptimal solutions are generated in the process of finding the optimal solution. If so desired, all feasible solutions with cost less than some specified amount can be generated. (This is important because not all pertinent data can be included in the object function. A suboptimal design may be the best overall design when other factors are considered.)

The major limitation in the method lies in determining the set of buses at which capacitor additions are to be considered. The solution is optimal only for the set of application buses specified and one set may result in a better solution than another. Several methods of determining the application buses will be discussed in Chapter 6.

Chapter 5

CAPACITOR ALLOCATION AS A PROBLEM IN DISCRETE OPTIMIZATION

In this chapter a new formulation of the capacitor allocation problem is presented. This formulation will take into account the inherent discrete nature of the variables involved. Section 5. 1 begins with a physical justification of the viewpoint to be taken. Section 5. 2 follows with the introduction of some convenient notation and a formulation of the problem in mathematical terms. A linearized version of the problem is then presented in Section 5. 3. Section 5. 4 concludes the chapter with a summary of the methods available for solving problems of discrete optimization.

5. 1 Classification of the Allocation Problem

As shown in the last chapter, the problem of shunt static capacitor allocation can be viewed as a constrained optimization problem. The goal is to satisfy system voltage constraints while minimizing the cost of doing so. The variables which can be manipulated to obtain this minimum are the locations of the installations and the amount of capacitance placed in them.

Power system design most frequently involves modification of an already existing network. This is the case with static capacitor additions. Capacitor banks are connected at substation buses, or, in the case of distribution networks, they are sometimes used on utility poles. This means that there are only a finite number of locations at which capacitors can be applied to the system. The location variable is therefore of a discrete nature and not continuously variable. Further discreteness is introduced into the problem by the fact that static capacitors are manufactured only in certain KVAR sizes. This means the amount of capacitance at a given location is also a discrete variable.

Problems of this nature are called by many different names in the mathematical literature. The discreteness of the variables leads to the term "discrete programming problem." Other terms which are commonly used are "integer programming problem" and "combinatorial optimization problem." The word "programming" here derives from the notion of defining a statement of actions or "program" to be followed in solving the problem.

5.2 General Mathematical Formulation

As a first step in formulating the capacitor allocation problem in mathematical terms it will perhaps be advantageous to establish some basic notation and terminology. The problem shall be based upon a specified set of buses, $A = \{\alpha_1, \alpha_2, \ldots, \alpha_M\}$ to be called the application set. These buses are to be the possible sites of capacitor installations. The α_I refer to the particular buses and

M is the total number of buses in the set. An allocation shall be defined as any installation of capacitor units at one or more of the buses in set A. Each particular allocation shall be specified by a column vector or array, $\bar{X}$, whose elements $X(\alpha_I)$ indicate the integer number of capacitor units connected at bus α_I. An allocation must always be defined relative to an array $\bar{B}$ which specifies the size of the capacitor units used at each bus. For example, the element $B(\alpha_I)$ is the susceptance of the unit size of capacitance which can be placed at bus α_I.

Now it is possible that in some particular allocation no capacitor units are installed at one or more of the buses in set A. In other words, one or more of the elements of the corresponding vector $\bar{X}$ are zero. Those buses which do have installations at them will be said to constitute the application subset, A_i, for that allocation. Note that, by definition, a bus included in a given A_i must have at least one unit of capacitance installed at it. Note also that there are 2^M-1 possible application subsets A_1, A_2, ..., A_{2^M-1}, in an application set of cardinality M.

Given a particular allocation, $\bar{X}$, it may be necessary for one or more of the capacitor banks specified by it to be switched out of the network during light load conditions. The fixed allocation remaining after the switched units have been disconnected from the network shall be indicated by the array $\overline{XF}$. $\overline{XF}$ is identical to $\bar{X}$ except that the elements corresponding to the switched units are zero. A fixed allocation defines a fixed subset of A_i denoted by A^f_{ij}, which contains the buses at which fixed banks have been placed. A switched subset denoted by A^s_{ij} can also be defined and is equal to $A_i - A^f_{ij}$. The fact that there are several switched or fixed subsets for each application subset A_i is indicated by the subscript j.

The vector of bus voltage magnitudes resulting from a specified allocation can be denoted as $\bar{V}(\bar{\xi}, \bar{B}, s_j)$ where $\bar{\xi}$ equals $\bar{X}$ or $\overline{XF}$ depending upon whether the switched units are connected to the system or not. s_j is an integer indicating one of a finite number of possible "states" of the system before capacitor additions. "State" here refers to the factors, such as generation, network configuration and loading, which collectively determine the initial voltage levels. (The states typically considered will represent the extremes in voltage levels anticipated.) It is understood that there exists some associated record which contains all of the information necessary to completely define a state, s_j.

Using this notation, the capacitor allocation problem for an N bus system can be put in the form of the following constrained optimization problem:

Minimize $\quad C = f(A_i, \bar{X}, \overline{XF}, \bar{B})$

subject to $\quad 0 \leq X(\alpha_I) \leq XM_0(\alpha_I)$ for $I = 1, 2, \ldots, M$ C1

$\bar{V}(\overline{XF}, \bar{B}, s_0) \leq \bar{V}_{max}$ C2

$\bar{V}(\bar{X}, \bar{B}, s_j) \geq \bar{V}^{s_j}_{min}$ C3

for $j=1, \ldots, L$

where

M is the total number of buses at which capacitors are to be applied. $(1 \leq M \leq N)$.

L is the total number of low voltage or outage conditions to be studied. Note: A single outage condition can consist of multiple equipment outages.

s_0 is the state corresponding to the highest voltage conditions expected.

s_j is the state corresponding to the j-th low voltage or outage case to be studied. $(j=1, \ldots, L)$.

$\bar{V}^{s_j}_{min}$ is an N-vector representing the lower limits on bus voltage magnitudes for system state s_j.

$\bar{V}_{max}$ is an N-vector representing the upper limits on bus voltages.

$f(A_i, \bar{X}, \overline{XF}, \bar{B})$ is a cost function which takes into account the cost of the capacitor units, the cost of any associated breakers and controls, and the cost of installation. That these costs may depend upon the particular application buses is reflected in the dependence upon A_i.

$XM(\alpha_I, s_j)$ is a limit placed on the number of capacitor units which can be placed at a bus α_I in addition to any already present for state s_j. This limit is set in order to limit voltage fluctuations during switching and also to increase reliability.

$$XM_0(\alpha_I) \quad = \min_{\forall s_j} XM(\alpha_I, s_j)$$

If $S(\alpha_I)$ is used to represent the set of all integers, $X(\alpha_I)$, which satisfy the constraint C1, the capacitor allocation problem can be stated as follows:

Find the allocation $(X(\alpha_1), X(\alpha_2), \ldots, X(\alpha_M))$, and the fixed allocation $(XF(\alpha_1), XF(\alpha_2), \ldots, XF(\alpha_M))$, in the product space $S(\alpha_1) \times S(\alpha_2) \times \ldots \times S(\alpha_M)$ which minimizes the function $f(A_i, \bar{X}, \overline{XF}, \bar{B})$ and satisfies the constraints C2 and C3.

Note that the cost of a switched capacitor bank of a given size will be more than the cost of an equivalent fixed bank due to the associated switches and controls. This is taken into account in the cost function $f(A_i, \bar{X}, \overline{XF}, \bar{B})$.

Note also that there are two variations of the problem just stated. One would involve the allocation of only fixed capacitor banks, the other the allocation of only switched capacitor banks. Their problem statements are similar except for the addition of the constraints $\overline{XF} = \bar{X}$ and $\overline{XF} = \bar{0}$ for the fixed and switched case respectively. The particulars of these variations of the problem will be discussed in Chapter 6.

The following definitions will be understood in what follows:

Definition:

A feasible solution (or solution) to a capacitor allocation problem specified relative to an application set A, shall be any allocation $\bar{X}$ and fixed allocation $\overline{XF}$ which satisfies constraints C1, C2 and C3.

Definition:

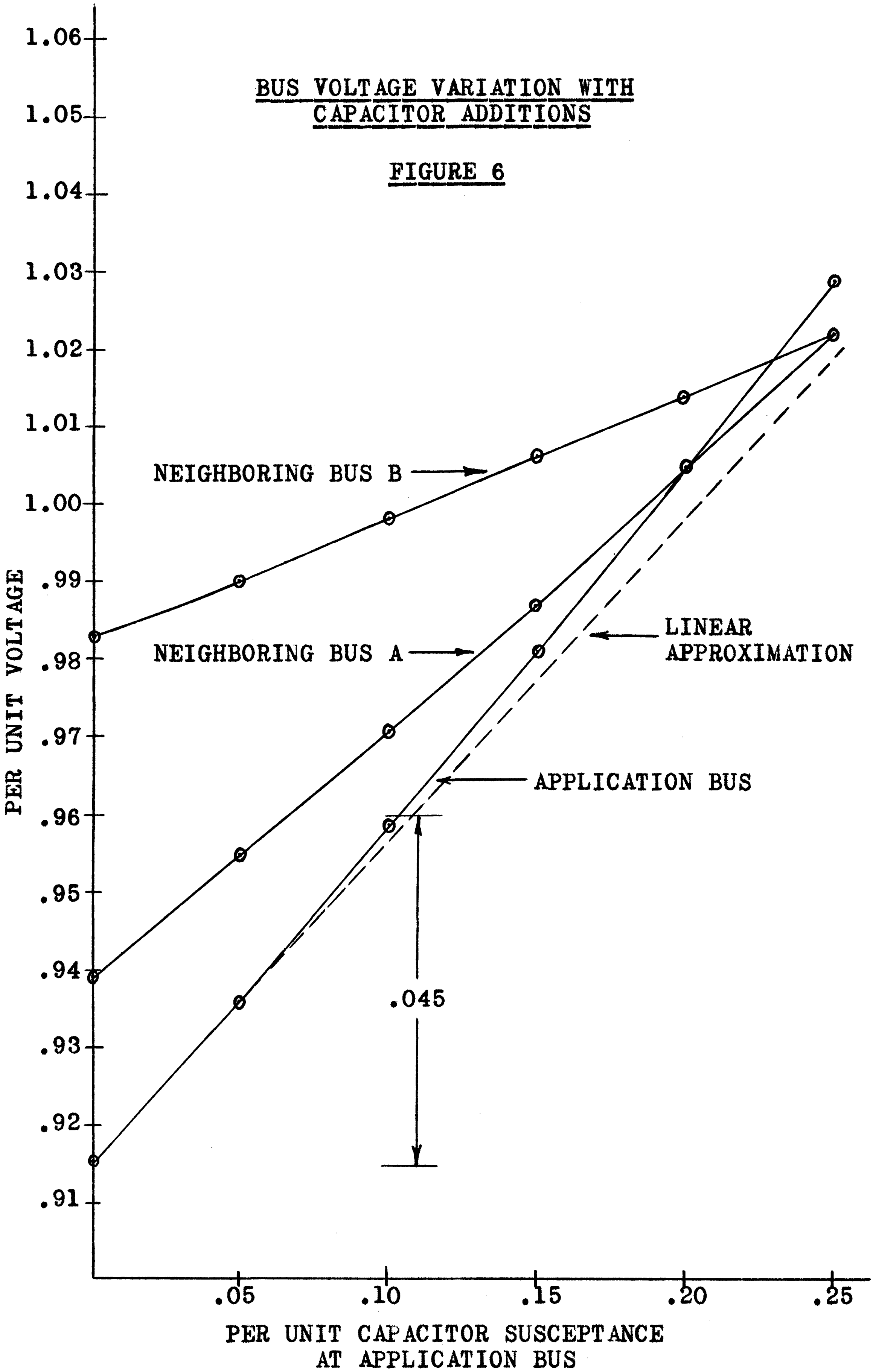
An optimal solution for such a problem shall be a feasible solution which minimizes the cost functional.

5.3 A Linear Formulation

In any method of solving the capacitor allocation problem some means must be available for computing the voltages resulting from a given allocation. The most accurate means available are the load flow analysis programs. But these are based on iterative solution techniques and may require a fairly large amount of time to converge to a solution when large systems of several hundred buses are involved. It would therefore be desirable if an alternative method, not based on successive iterations, could be found—even if it gave somewhat more approximate results. One's first thought is to examine plots of bus voltage change with shunt capacitor susceptance at a single bus in the hope of obtaining a clue to an appropriate model. This was done for plots obtained from the Consumers Power Company Alma-Midland and Flint 46Kv subtransmission networks.[†] Figure 6 illustrates the typical variation found. The curves are nonlinear, but not decidedly so. And, as can be seen from the lower

----

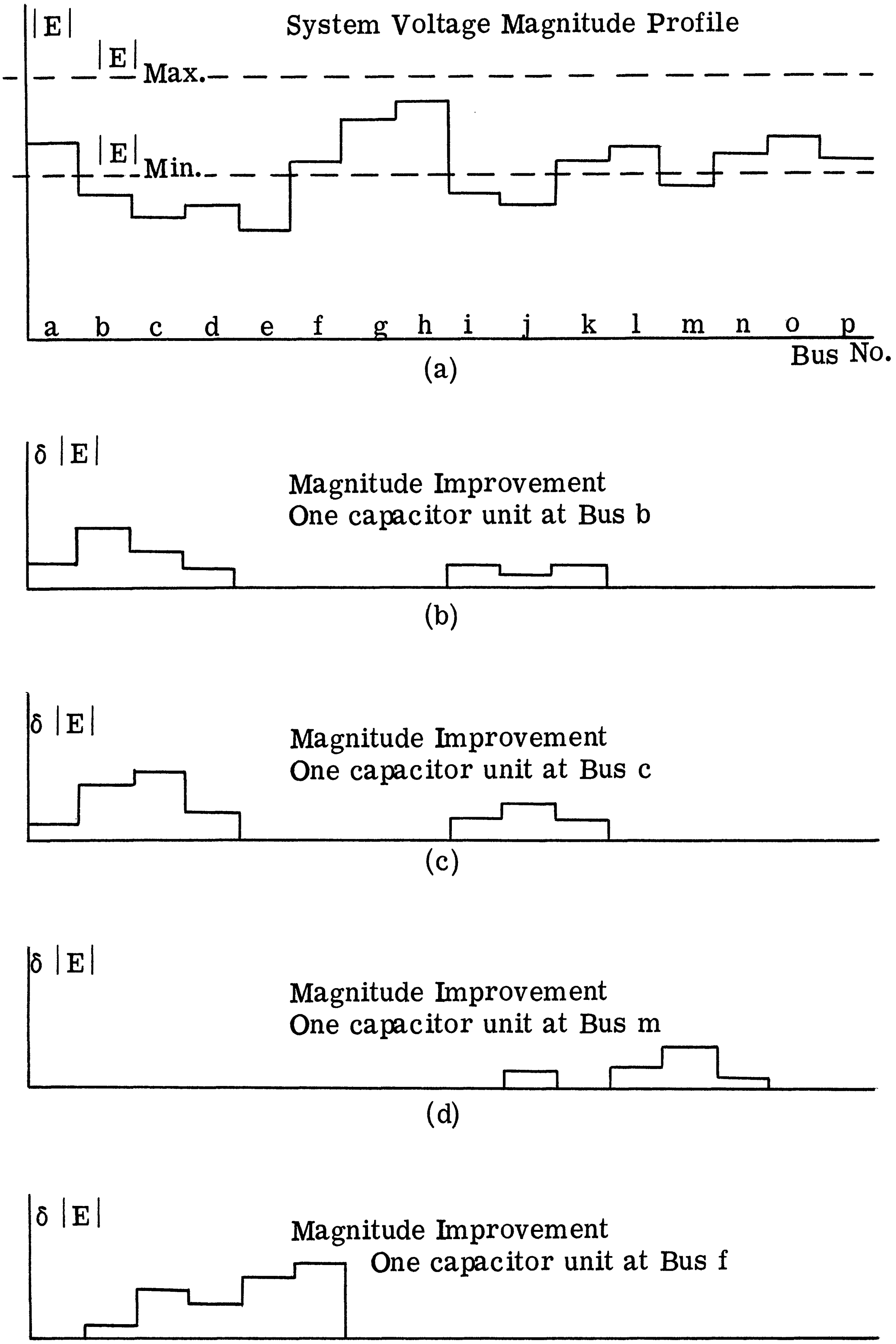
[†] These plots were obtained from the load flow analysis program described in the Appendix. It does not contain some of the features found in many modern large scale load flow programs but its accuracy should be sufficient for many planning studies. (See page 150). More will be said about the importance of system model accuracy in Section 8.3.


BUS VOLTAGE VARIATION WITH CAPACITOR ADDITIONS
FIGURE 6
1.06
1.05
1.04
1.03
1.02
1.01
1.00
.99
.98
.97
.96
.95
.94
.93
.92
.91
PER UNIT VOLTAGE
NEIGHBORING BUS B
NEIGHBORING BUS A
LINEAR APPROXIMATION
APPLICATION BUS
.045
.05
.10
.15
.20
.25
PER UNIT CAPACITOR SUSCEPTANCE AT APPLICATION BUS

curve, a linear extrapolation based on the first unit of capacitance applied is a fairly good approximation to the curve over a limited range in voltage increase. Since, in many capacitor appplications the voltage increases produced by a given bank are limited to a small percentage (say 4.5%) of the nominal voltage, these graphs would seem to indicate that a linear model for the voltage changes with capacitor susceptance should be adequate for engineering purposes. However, the real verification of this hypothesis can only be obtained by comparison of linear model voltage predictions with load flow results in cases where capacitors are applied at many of the system buses. This has been done and the accuracy of the linear model has proven adequate for most purposes. More will be said about this in a later chapter.

A graphical interpretation of the problem, as based on a linear model, is easily visualized. Consider the problem of allocating fixed capacitors on a network of arbitrary voltage level. Figure 7a shows how the voltage profile of such a network might appear if plotted in a bar graph. It is necessary that all of the bus voltages be maintained above the level indicated by V_{min} and below the level indicated by V_{max}. In a typical problem two or more bar graphs such as the one in Figure 7a will be considered—one for state s_0, where the constraint V_{max} is important, and one or more for the outage or low voltage states s_j, where V_{min} is important.

The change in system voltage magnitude resulting from the



(a)

(b)

(c)

(d)

(e)

PACKAGING ANALOGY
Figure 7

application of a single unit of capacitance at a given bus can also be plotted in a bar graph. Figures 7b, 7c, 7d, and 7e show how such graphs might appear. Each of the graphs can be thought of as "packages" containing "building blocks" of voltage improvement. Each package has a cost associated with it. No package can be split into smaller packages containing only some of the improvement blocks. Every voltage change associated with a given capacitor addition must be accepted. Note also, that the package associated with a given capacitor installation will in general be different for each state, s_j, considered. This is because network configuration, loads and generation will determine the amount and locations of the resulting voltage rises. In this analogy, the problem is to purchase packages of voltage improvement blocks, which when stacked onto the voltage profiles being considered, will satisfy the voltage constraints at the minimum cost.

Assuming a linear voltage change model, the constrained optimization problem appears as follows:

Minimize $$C = f(A_i, \bar{X}, \overline{XF}, \bar{B})$$

subject to $$0 \leq X(\alpha_I) \leq XM_0(\alpha_I) \quad \text{for } I=1,\ldots,M \qquad \text{C1}$$

$$\bar{V}^o_{s_o} + D_{s_o}\overline{XF} \leq \bar{V}_{max} \qquad \text{LC2}$$

$$\bar{V}^o_{s_j} + D_{s_j}\bar{X} \geq \bar{V}^{s_j}_{min} \qquad \text{LC3}$$

for j=1, . . . , L

where

D_{s_j} is an N × M matrix whose p-th column consists of the change in voltage magnitude at each network bus resulting from the application of one unit of shunt capacitance of susceptance $B(\alpha_p)$ at the p-th bus in the application set when the system is in state s_j. $(j=0, 1, \ldots, L)$.

$\bar{V}^o_{s_j}$ is an N-vector of the existing system voltages under state s_j. $(j=0, 1, \ldots, L)$.

The remaining notation is identical to that presented earlier.

The similarity between the linear model used here and the model of the "packaging analogy" is obvious. Each column of a D_s-matrix[†] corresponds to one of the packages of voltage improvement. Here it is recognized that the packages obtained from a given capacitor installation will depend upon the state of the system.

Strictly speaking, each package of voltage improvement is a vector of complex quantities as each bus voltage is represented by a real and an imaginary part. In the formulation above it was assumed that they were vectors of scalar quantities which could be added algebraically to the existing voltage magnitude vector. A more accurate model would involve complex addition and the constraint equations would appear as follows:

[†] The D_{s_j}-matrices shall be referred to as D_s-matrices.

$$|\bar{E}^{o}_{s_o} + D_{s_o}\bar{X}| \le \bar{V}_{max} \qquad CC2$$

$$|\bar{E}^{o}_{s_j} + D_{s_j}\bar{X}| \ge \bar{V}^{s_j}_{min} \qquad \text{for } (j=1,\ldots,L) \qquad CC3$$

where $E^{o}_{s_i}$ is an N-vector of complex bus voltages and the columns of the D_s-matrices are the complex voltage changes under capacitor additions.

As already mentioned, it will be shown in a later chapter that the scalar model is sufficiently accurate. Of course the scalar model has the advantage of requiring less computation and less storage space in the computer than the complex linear model.

5.4 Solution Methods for Discrete Optimization Problems

Now that the capacitor allocation problem has been formulated as a discrete optimization or integer programming problem, it is appropriate to discuss the techniques available for solving problems of this type. Of the numerous methods described in the literature, some have met with reasonable computational success and others have not. Each has its own advantages and disadvantages. In this case, the requirements of the capacitor allocation problem help to narrow the choice of possible methods.

A basic method common to many approaches to solving integer programming problems with linear constraint equations and cost functional is to solve a sequence of linear programming problems [2, 3]. If at any stage a noninteger solution is obtained, supplementary linear constraints are deduced which must be satisfied by any feasible integer solution. The successive addition of these constraints "transforms" the linear program into one whose solution has the desired integer property. The difficulty with these, and many other available algorithms, is that they are "dual methods" and no feasible solution to the problem of interest is obtained until the optimal solution is found. Since a list of the "best" feasible solutions (containing the optimum) may be desired, these methods are not ideal. In addition it may be discovered in the middle of the computation process that computation time is becoming excessive. If it is decided to terminate the procedure prematurely for this reason, one is left without any (even suboptimal) solution to show for the computational effort already expended.

One promising class of methods involves partitioning the problem space into regions and then examining each region to see if the necessary conditions for a feasible solution (or optimal) solution are satisfied there [20]. If they are satisfied, the region is partitioned into still smaller regions which are in turn examined. This subpartitioning is repeated until a feasible solution is obtained. Then the search is directed to the discovery of better and better feasible solutions until the optimum solution is obtained. One necessary condition might involve a lower bound on the cost function in a given region. A region can be excluded from further consideration when this bound exceeds the cost of some known feasible solution. One important property of this approach is that premature termination of the solution process because of excessive execution time, may still leave one with a usuable, although suboptimal, solution. It also has the desirable property of providing suboptimal solutions as a by-product of the search process and of providing the opportunity to search for suboptimal solutions if so desired. Suboptimal solution search is accomplished by examing some of the regions that would otherwise be rejected.

The principal disadvantage of partitioning methods is that computationally they may be quite unpredictable in length. In addition, the information which must be retained during the repeated partitioning operations may place severe demands upon the computer memory available. Power system analysis methods already tax computer memory with the large amount of system data which must be stored. Therefore this limitation of the partitioning methods is an important one—particularly to the propsective user with a small computer system.

Methods which utilize a controlled enumerative procedure which implicitly considers all potential solutions to the problem are quite common in the literature [21, 22, 31]. Among these methods are those referred to as backtrack programming, combinatorial programming, partial enumeration and implicit enumeration. Basically, they all proceed in some systematic manner through the sample vectors in the problem space. Decision rules are used whereby whole sequences of vectors which cannot possibly contain a solution are skipped over. Again, the search is directed first to the discovery of a feasible solution and then to successively better feasible solutions until the optimal solution is obtained. Thus these methods share the desirable properties of the partitioning method.

It has been suggested that programming methods, such as the ones discussed in the last paragraph, are based more on art than on science. Their effectiveness depends to a large extent upon the discovery of appropriate properties of the problem which can be used to eliminate certain sample vectors from consideration with a minimum of effort. However, this observation in no way detracts from the usefulness of the method.

"Heuristic" techniques have proved to be valuable in instances where problems are hopelessly large computationally [35]. They usually function as a means of removing from the set of alternatives to be examined those which are most likely not good solutions. Obviously, they are not claimed to be able to find optimal solutions as they may actually discard them in the "weeding out" process. However they do aid in finding good solutions when all else fails and they may at times be used to direct more precise procedures in an attempt at making them more efficient. An example of this latter use is in the algorithm to be described in the next chapter.

Chapter 6

THE IMPLICIT ENUMERATION ALGORITHMS

In this chapter algorithms are developed to solve the capacitor allocation problem as formulated in Chapter 5. Section 6. 1 provides an introduction to the method to be used. Then an algorithm for fixed unit allocation is developed in Sections 6. 2 through 6. 5. This is extended for use in switched and simultaneous fixed and switched unit allocation in Sections 6. 6 and 6. 7 respectively. Section 6. 8 points out the flexibility of the algorithms derived.

6. 1 Introduction

The first part of this chapter will center around a means of solving a subproblem of the capacitor allocation problem—namely the allocation of fixed capacitor banks. Later in the chapter the techniques developed for fixed capacitor allocation will be extended to the cases where all banks are switched and where banks can be either fixed or switched. In some respects, the allocation of fixed capacitors is a more interesting problem than the allocation of switched banks. This is because, even if the low voltage constraints can be satisfied with capacitor additions, a solution may not exist with fixed capacitors which will also satisfy the high voltage constraint under normal, light load conditions. When all capacitors can be removed from the network under these conditions, there is usually no danger of violating the high voltage constraints.

For the first part of the chapter, then, the problem to be considered can be stated as follows:

Find the allocation $\bar{X}$ in the product space $S(\alpha_1) \times S(\alpha_2) \times \ldots \times S(\alpha_M)$ which with $\overline{XF} = \bar{\bar{X}}$ minimizes the function $f(A_i, \bar{X}, \overline{XF}, \bar{B})$[†] and satisfies the constraints C2 and C3.[††]

The solution method to be used is basically an enumeration scheme with decision rules which enable some of the allocations in the enumeration sequence to be skipped over and ignored. It is based in part on an algorithm described by Lawler and Bell [22]. Since many of the possible allocations are only implicitly (not explicitly) enumerated, the algorithm shall be referred to as an implicit enumeration algorithm (IEA). The explicit enumeration portion of the process is first directed to the discovery of a feasible solution. Once this has been obtained, only allocations with the possibility of having less than or equal cost are explicitly enumerated. Therefore the algorithm has the desirable property of providing suboptimal solutions as a by-product. The opportunity to search directly for suboptimal solutions is also available. A further advantage of IEA is that the underlying enumeration scheme in itself requires little computer storage for procedural planning and control. Some other methods require extensive bookkeeping to record the possibilities already evaluated

----

[†] Or $f(A_i, \bar{X}, \bar{X}, \bar{B})$

[††] See page 54 .

and to select the next possible solutions to consider. These statements will be elaborated upon in the following sections. The next section describes an important property of the cost and voltage constraint equations that enables implicit enumeration to take place.

6.2 Important Properties of the Constraint and Cost Equations

Under the conditions encountered on electrical power systems, the bus voltages do not decrease with the addition of shunt capacitors. One can argue that, theoretically at least, situations could exist in which this would not be the case. Fortunately, such situations rarely, if ever, arise in practical cases. Under normal conditions then, each element of the bus voltage vector, $\bar{V}(\bar{X}, \bar{B}, s_j)$, is <u>monotone nondecreasing</u> in each of the elements of $\bar{X}$. (This means that the elements of each D_s-matrix are greater than or equal to zero.) This property plays an important role in the Implicit Enumeration Algorithm.

The behavior of the cost functional $C = f(A_i, \bar{X}, \bar{X}, \bar{B})$ is also important to the algorithm's operation. Up to this point, discussion of the cost functional has been purposely general. From the standpoint of the algorithm, its exact structure is not important. What is important is that it must be <u>monotone nondecreasing</u> in each of the elements of $\bar{X}$. That this is the case for any practical cost function is a fairly safe assumption to make because with each capacitor addition the cost function must increase by at least the cost of the capacitor unit added. In actuality, the assumption that the cost function is strictly increasing in each element of $\bar{X}$ is justified. If things such as labor and switchgear costs, etc. are included in the cost functional, this would not affect its strictly increasing property.

6.3 The Explicit Enumeration Scheme

The enumeration scheme underlying the implicit enumeration algorithm systematically generates every allocation $\bar{X}$, in the problem space $S(\alpha_1) \times S(\alpha_2) \times \ldots \times S(\alpha_M)$. The only deviations in the enumeration sequence occur in instances where upcoming allocations in the sequence can be shown to be unacceptable without explicitly considering them. In such cases these allocations are skipped over and the enumeration scheme recommences as if they had, in fact, been considered. The structure of the enumeration process is based upon a partitioning of the vectors in the problem product space into smaller sets of vectors. One of the partitions is selected and the vectors within it are enumerated (at least implicitly). Then another partition is selected and the enumeration process is repeated. This is continued until every possible vector has been considered, either implicitly or explicitly. Before examining the exact means by which this partitioning and enumeration is accomplished, it will be convenient to introduce some additional notation.

Each allocation problem is defined with respect to an application set of buses, A, ɛt which capacitor units may be applied. For an application set of cardinality M, there are 2^M-1 application subsets,

$A_1, A_2, \ldots, A_{2^M-1}$. Three things can be associated with each A_i: a product space P_i, and two special vectors denoted by $\bar{X}^i{}_{max}$ and $\bar{X}^i{}_{min}$. P_i shall be defined as the product space $P_i(\alpha_1) \times P_i(\alpha_2) \times \ldots \times P_i(\alpha_M)$ where $P_i(\alpha_I) = \{0\}$ if $\alpha_I \notin A_i$ and $P_i(\alpha_I) = \{1, \ldots, XM_0(\alpha_I)\} \overset{\Delta}{=} S^*(\alpha_I)$ if $\alpha_I \in A_i$. Note that the difference between $S^*(\alpha_I)$ and $S(\alpha_I)$ is the absence of the element zero. It is easily seen that one of the vectors in P_i is a "maximal" vector in the sense that its elements are all greater than or equal to the corresponding elements of any other vector in P_i. Similarly, there exists a minimal vector in P_i. These extremal elements are as follows:

$$\bar{X}^i{}_{max} = \text{col}[X^i{}_{max}(\alpha_1), X^i{}_{max}(\alpha_2), \ldots, X^i{}_{max}(\alpha_M)] \text{ where}$$
$$X^i_{max}(\alpha_I) = 0 \text{ if } \alpha_I \notin A_i \text{ and } X^i_{max}(\alpha_I) = XM_0(\alpha_I) \text{ if } \alpha_I \in A_i.$$
$$\bar{X}^i{}_{min} = \text{col}[X^i{}_{min}(\alpha_1), X^i{}_{min}(\alpha_2), \ldots, X^i{}_{min}(\alpha_M)] \text{ where}$$
$$X^i_{min}(\alpha_I) = 0 \text{ if } \alpha_I \notin A_i \text{ and } X^i_{min}(\alpha_I) = 1 \text{ if } \alpha_I \in A_i.$$

Note that the 2^M-1 product spaces P_i, associated with A, form a disjoint covering of the problem product space $S(\alpha_1) \times S(\alpha_2) \times \ldots \times S(\alpha_M)$. This is equivalent to saying that the vector spaces P_i form a partitioning of the product space $S(\alpha_1) \times S(\alpha_2) \times \ldots \times S(\alpha_M)$. In the enumeration scheme, these partitions are considered in an order determined by the cardinality of their associated application subset, A_i. The sequence is such that all the P_i associated with subsets containing one element occur first, then those associated with subsets of two elements, etc. The last vector set to be considered is the one associated with the application subset $A_{2^M-1} = A$. To clarify this, the sequence of A_i and P_i used is shown in Figure 8. The corresponding sequence of $\bar{X}^i_{max}$ and $\bar{X}^i_{min}$ is shown in Figure 9.

Enumeration of the vectors within a given product space (or partition) P_i is accomplished as follows: The first vector $\bar{X}$ is identical to the vector $\bar{X}^i_{min}$. The generation of each additional vector is initiated by adding 1 to $X(\alpha_M)$. If the resulting $X(\alpha_M) \leq X^i_{max}(\alpha_M)$ then no "carry" is made and the resulting vector is the next in the sequence. However, if $X(\alpha_M) + 1 > X^i_{max}(\alpha_M)$, then $X(\alpha_M)$ is set equal to 1 if $X^i_{max}(\alpha_M) \neq 0$ and equal to 0 otherwise. The "carry" is continued by adding 1 to $X(\alpha_{M-1})$. Again the resulting $X(\alpha_{M-1})$ is checked to see if it is less than or equal to $X^i_{max}(\alpha_{M-1})$. If it is, the resulting vector is the next in the sequence. Otherwise $X(\alpha_{M-1})$ is set to 1 or 0 as in the case of $X(\alpha_M)$ and 1 is added or "carried" to $X(\alpha_{M-2})$. This "carry" operation is continued, with the subscript of α decreasing each time, until $X(\alpha_J) + 1 \leq X^i_{max}(\alpha_J)$ for some J. All $X(\alpha_I)$ for which $I > J$ have been set to 1 if $X^i_{max}(\alpha_I) \neq 0$ and set to zero otherwise. When the condition $X(\alpha_1) + 1 > X^i_{max}(\alpha_1)$ occurs during a carry, every vector in the product space P_i has been generated. Figure 10 shows an example of a vector sequence resulting from this process. In later sections this process of adding 1 to $X(\alpha_M)$ and performing all carries will be denoted by $\bar{X} \leftarrow \bar{X} + 1$.

In summary, the sequence of allocations $\bar{X}$, generated by the underlying enumeration scheme consists first of the vectors in P_1,

A_i-SUBSET SEQUENCE	P_i-PRODUCT SPACE SEQUENCE
$A_1 = \{\alpha_1\}$	$S^*(\alpha_1) \times \{0\} \times \{0\} \times \ldots\ldots\ldots\ldots \times \{0\}$
$A_2 = \{\alpha_2\}$	$\{0\} \times S^*(\alpha_2) \times \{0\} \times \ldots\ldots\ldots\ldots \times \{0\}$
$\ldots\ldots\ldots$	$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$
$A_M = \{\alpha_M\}$	$\{0\} \times \{0\} \times \ldots\ldots\ldots\ldots \times \{0\} \times S^*(\alpha_M)$
$A_{M+1} = \{\alpha_1, \alpha_2\}$	$S^*(\alpha_1) \times S^*(\alpha_2) \times \{0\} \times \ldots\ldots\ldots \times \{0\} \times \{0\}$
$A_{M+2} = \{\alpha_1, \alpha_3\}$	$S\times(\alpha_1) \times \{0\} \times S^*(\alpha_3) \times \{0\} \times \ldots\ldots \times \{0\} \times \{0\}$
$\ldots\ldots\ldots\ldots\ldots$	$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$
$A_K = \{\alpha_{M-1}, \alpha_M\}$	$\{0\} \times \ldots\ldots\ldots\ldots \times \{0\} \times S^*(\alpha_{M-1}) \times S^*(\alpha_M)$
$A_{K+1} = \{\alpha_1, \alpha_2, \alpha_3\}$	$S^*(\alpha_1) \times S^*(\alpha_2) \times S^*(\alpha_3) \times \{0\} \times \ldots\ldots\ldots \times \{0\}$
$A_{K+2} = \{\alpha_1, \alpha_2, \alpha_4\}$	$S^*(\alpha_1) \times S(\alpha_2) \times \{0\} \times S^*(\alpha_4) \times \{0\} \times \ldots\ldots \times \{0\}$
$\ldots\ldots\ldots\ldots\ldots$	$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$
$A_{2^M-2} = \{\alpha_2, \alpha_3, \ldots, \alpha_M\}$	$\{0\} \times S^*(\alpha_2) \times S^*(\alpha_3) \times \ldots\ldots\ldots\ldots \times S^*(\alpha_M)$
$A_{2^M-1} = \{\alpha_1, \alpha_2, \ldots, \alpha_M\}$	$S^*(\alpha_1) \times S^*(\alpha_2) \times S^*(\alpha_3) \times \ldots\ldots\ldots\ldots \times S^*(\alpha_M)$

Figure 8. A_i AND P_i SEQUENCES

$\bar{X}^{i}_{min}$-VECTOR SEQUENCE	$\bar{X}^{i}_{max}$-VECTOR SEQUENCE
$(1, 0, \ldots\ldots\ldots, 0)$	$(XM_0(\alpha_1), 0, \ldots\ldots\ldots\ldots\ldots, 0)$
$(0, 1, 0, \ldots\ldots\ldots, 0)$	$(0, XM_0(\alpha_2), 0, \ldots\ldots\ldots\ldots\ldots, 0)$
$\ldots\ldots\ldots\ldots\ldots$	$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$
$(0, \ldots\ldots\ldots, 0, 1)$	$(0, \ldots\ldots\ldots\ldots\ldots 0, XM_0(\alpha_M))$
$(1, 1, 0, \ldots\ldots\ldots, 0)$	$(XM_0(\alpha_1), XM_0(\alpha_2), 0, \ldots\ldots\ldots\ldots, 0)$
$(1, 0, 1, 0, \ldots\ldots, 0)$	$(XM_0(\alpha_1), 0, XM_0(\alpha_2), 0, \ldots\ldots\ldots\ldots, 0)$
$\ldots\ldots\ldots\ldots\ldots$	$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$
$(0, \ldots\ldots\ldots, 0, 1, 1)$	$(0, \ldots\ldots\ldots\ldots, 0, XM_0(\alpha_{M-1}), XM_0(\alpha_M)$
$(1, 1, 1, 0, \ldots\ldots, 0)$	$(XM_0(\alpha_1), XM_0(\alpha_2), XM_0(\alpha_3), 0, \ldots\ldots\ldots, 0)$
$(1, 1, 0, 1, 0, \ldots\ldots, 0)$	$(XM_0(\alpha 1), XM_0(\alpha_2), 0, XM_0(\alpha_4), 0, \ldots\ldots\ldots, 0)$
$\ldots\ldots\ldots\ldots\ldots$	$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$
$(0, 1, 1, \ldots\ldots\ldots, 1)$	$(0, XM_0(\alpha_1), \ldots\ldots\ldots\ldots, XM_0(\alpha_M))$
$(1, 1, \ldots\ldots\ldots, 1)$	$(XM_0(\alpha_1), \ldots\ldots\ldots\ldots\ldots, XM_0(\alpha_M))$

Figure 9. EXTREMAL VECTOR SEQUENCES

Figure 10

Example of the $\bar{X}$-Vector Sequence

$$\bar{X}^{i}_{max} = (3, 2, 0, 2, 0)$$

$\bar{X}$-VECTOR

(1, 1, 0, 1, 0)

(1, 1, 0, 2, 0)

(1, 2, 0, 1, 0)

(1, 2, 0, 2, 0)

(2, 1, 0, 1, 0)

(2, 1, 0, 2, 0)

(2, 2, 0, 1, 0)

(2, 2, 0, 2, 0)

(3, 1, 0, 1, 0)

(3, 1, 0, 2, 0)

(3, 2, 0, 1, 0)

(3, 2, 0, 2, 0)

OVERFLOW

then those in P_2, P_3, etc. up to P_{2^M-1}. The order of occurrence of the vectors within a given P_i is determined by the procedure described in the last paragraph.

6.4 Implicit Enumeration Schemes

In this section means will be discussed of implicitly enumerating or skipping over sub-sequences of vectors in the explicit enumeration scheme. The first schemes to be discussed will involve means of justifying the omission of all of the vectors in a given product space, P_i. These will be given as a series of "rules."

Rule 1:

If $\bar{V}(\bar{X}^i_{max}, \bar{B}, s_j) < \bar{V}^{s_j}_{min}$ for any system state, s_j=1, . . . , L, then all the vectors in product space P_i are skipped over and those in the next product space in the sequence are considered. (Note: This could happen for state s_0.)

Justification:

Since each element in the vector valued function $\bar{V}(\bar{X}, \bar{B}, s_j)$ is monotone nondecreasing in each element of $\bar{X}$, each element of $\bar{V}(\bar{X}, \bar{B}, s_j)$ is at its maximum for a given product space P_i when $\bar{X} = \bar{X}^i_{max}$. If $\bar{V}(\bar{X}^i_{max}, \bar{B}, s_j)$ doesn't satisfy the low voltage constraints, no vector in P_i can.

Rule 2:

If C' is a cost above which all more costly capacitor allocations are to be rejected, then if $f(A_i, \bar{X}^i_{min}, \bar{X}^i_{min}, \bar{B}) > C'$ all vectors in the product space P_i are skipped over and those in the next product space in the sequence are considered.

Justification:

Because of the monotone nondecreasing property of the cost function, the least costly vector in a product space P_i is $\bar{X}^i_{min}$.

Rule 3:

If the least expensive capacitor allocation is desired and an allocation with cost C* has been obtained, then the vectors in the product space P_i are skipped over and those in the next in sequence considered if $f(A_i, \bar{X}^i_{min}, \bar{X}^i_{min}, \bar{B}) > C^*$.

Justification:

The justification is identical to that for Rule 2.

What shall be known as the SKIP OPERATION will now be defined. Immediately following it, some rules will be presented whereby sub-sequences of vectors within a product space can be skipped using the SKIP OPERATION.

SKIP OPERATION

The first element $X(\alpha_J)$ from the α_M-end of the $\bar{X}$-vector which is not a zero and not a one is set to 1 and $X(\alpha_{J-1})$ is increased by 1. All necessary carries are then performed.

The vectors following an arbitrary $\bar{X}$ in the enumeration sequence will either be greater than or noncomparable to $\bar{X}$ by the relation $\leq$.[†]

[†] $\bar{X} \leq \bar{Y}$ if and only if $\bar{X}_i \leq \bar{Y}_i$ for all i. $\bar{X}$ and $\bar{Y}$ are noncomparable if $\bar{X} \not\leq \bar{Y}$ and $\bar{X} \not\geq \bar{Y}$.

Figure 11

Examples of the Skip Operation

$$\overline{XMAX}_i = (2, 3, 0, 2, 3)$$

$$\bar{X} = (1, 2, 0, 1, 1) \xrightarrow{\text{SKIP OPERATION}} \bar{X} = (2, 1, 0, 1, 1)$$

VECTORS SKIPPED

(1, 2, 0, 1, 2)
(1, 2, 0, 1, 3)
(1, 2, 0, 2, 1)
(1, 2, 0, 2, 2)
(1, 2, 0, 2, 3)
(1, 3, 0, 1, 1)
(1, 3, 0, 1, 2)
(1, 3, 0, 1, 3)
(1, 3, 0, 2, 1)
(1, 3, 0, 2, 2)
(1, 3, 0, 2, 3)

$$\bar{X} = (1, 1, 0, 1, 3) \xrightarrow{\text{SKIP OPERATION}} \bar{X} = (1, 1, 0, 2, 1)$$

VECTORS SKIPPED
NONE

$$\bar{X} = (1, 2, 0, 2, 1) \xrightarrow{\text{SKIP OPERATION}} \bar{X} = (1, 3, 0, 1, 1)$$

VECTORS SKIPPED

(1, 2, 0, 2, 2)
(1, 2, 0, 2, 3)

The vector generated by the SKIP OPERATION is the first noncomparable vector following $\bar{X}$ in the sequence. As shown in Figure 11, each vector element in the sub-sequence skipped, is greater than or equal to the corresponding element in the initial $\bar{X}$. With this in mind several additional rules can be stated.

Rule 4:

If $\bar{V}(\bar{X}, \bar{B}, s_0) > \bar{V}_{max}$ then apply the SKIP OPERATION to $\bar{X}$ and resume the enumeration sequence with the resulting vector. If the operation skips the remaining vectors in the present vector set, proceed to the next vector set in the sequence.

Justification:

Because of the monotone non-decreasing property of $\bar{V}(\bar{X}, \bar{B}, s)$, the next vector in the explicit enumeration sequence which can possibly satisfy the high voltage bound is the one produced by the SKIP OPERATION.

Rule 5:

If $f(A_i, \bar{X}, \bar{X}, \bar{B}) > C'$ (or C^*),[†] then apply the skip operation to $\bar{X}$ and resume the enumeration sequence with the resulting vector. If the operation skips the remaining vectors in the present vector set, proceed to the next vector set in the sequence.

[†] C' and C^* are used as defined in Rules 2 and 3.

Justification:

Because of the monotone non-decreasing property of $f(A_i, \bar{X}, \bar{X}, \bar{B})$ the next vector in the explicit enumeration sequence which can possibly have less cost than the present $\bar{X}$, is the one produced by the SKIP OPERATION.

Rule 6:

If $\bar{X}$ is a feasible solution, then apply the SKIP OPERATION to it and resume the enumeration sequence with the resulting vector. If necessary, proceed to the next vector set in the sequence.

Justification:

Each of the skipped vectors represents further capacitor additions at buses which already have satisfactory voltage levels. They only represent allocations with higher cost and voltage levels than the present allocation, $\bar{X}$. They therefore provide no further useful information.

Note that if the SKIP OPERATION is applied to $\bar{X}^i_{min}$ all of the vectors in the product space P_i are skipped over.

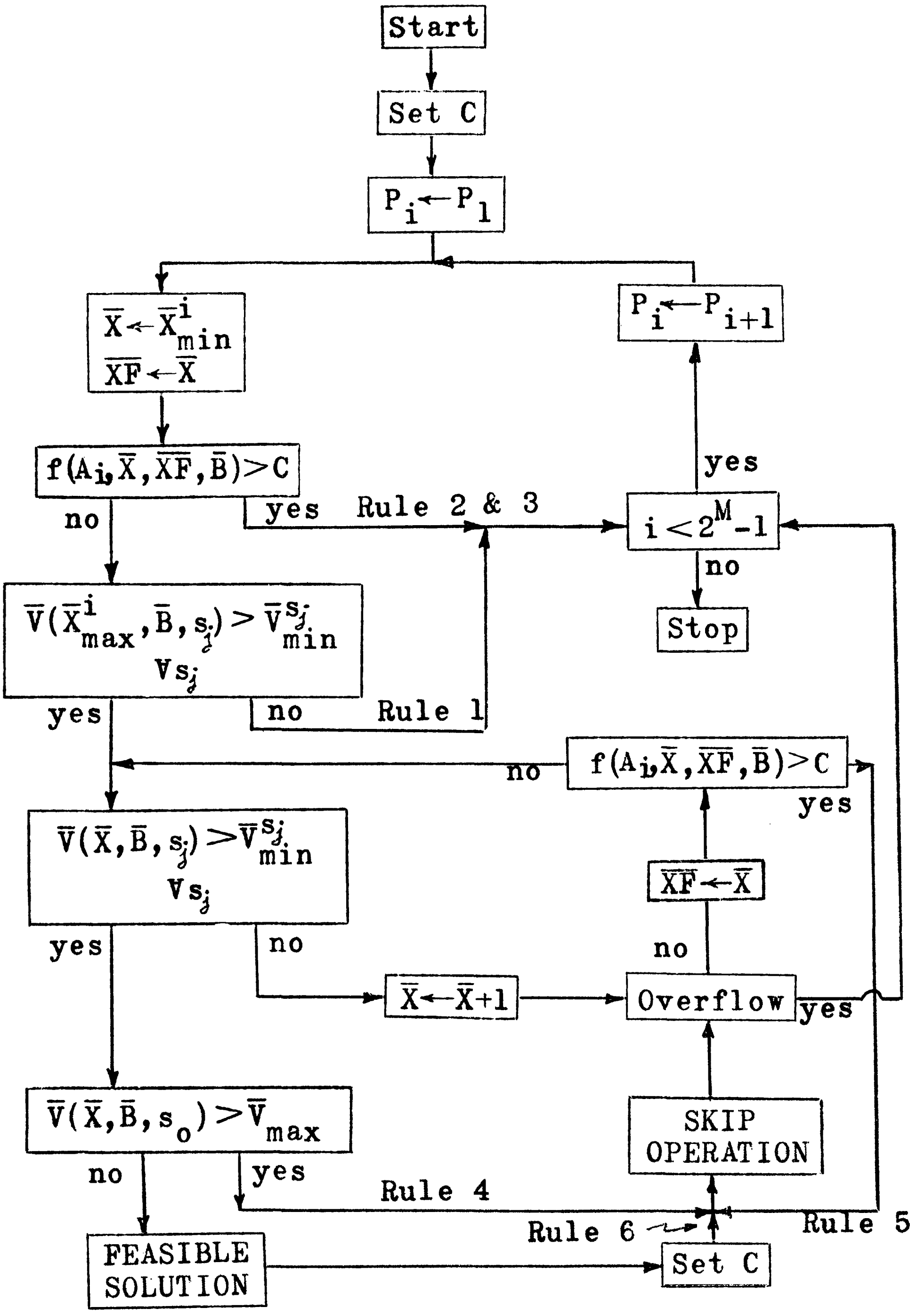
The above rules provide the means whereby whole sub-sequences of vectors in the explicit enumeration sequence can be skipped over or "implicitly enumerated." The next section describes how they are incorporated into an algorithm for fixed capacitor allocation.

6.5 Fixed Capacitor Allocation Algorithm

Figure 12 shows a flow diagram of an implicit enumeration algorithm as applied to fixed capacitor allocation. For the most part, it should be self-explanatory in view of the discussion in the last few sections. There are two boxes containing the instruction "Set C" — one at the top and one at the bottom of the diagram. The operations performed by these instructions will depend upon what the user of the algorithm is trying to accomplish. If his primary interest is in obtaining the optimum solution, then C is set to some arbitrary large member in the first (top) box. This number must be larger than the cost of any anticipated solution to the problem. The operation of the second (bottom) box is then to set C equal to the cost of the feasible solution just obtained. In this manner, allocations with cost greater than the lowest cost feasible solution obtained so far, will be skipped over. On the other hand, if the user wishes to obtain all feasible solutions with cost less than some specified amount, then C is set to this amount in the first "set C" operation and the second "set C" operation does nothing to this value.

Most of the notation used on the diagram has been introduced in the preceding sections. $P_i \leftarrow P_1$ means "place the product space P_i under consideration" and $P_i \leftarrow P_{i+1}$ means "consider the next product space in the sequence." It should be remembered that there are 2^M-1 such product spaces. The operation $\bar{X} \leftarrow \bar{X}+1$ was defined on page 72.

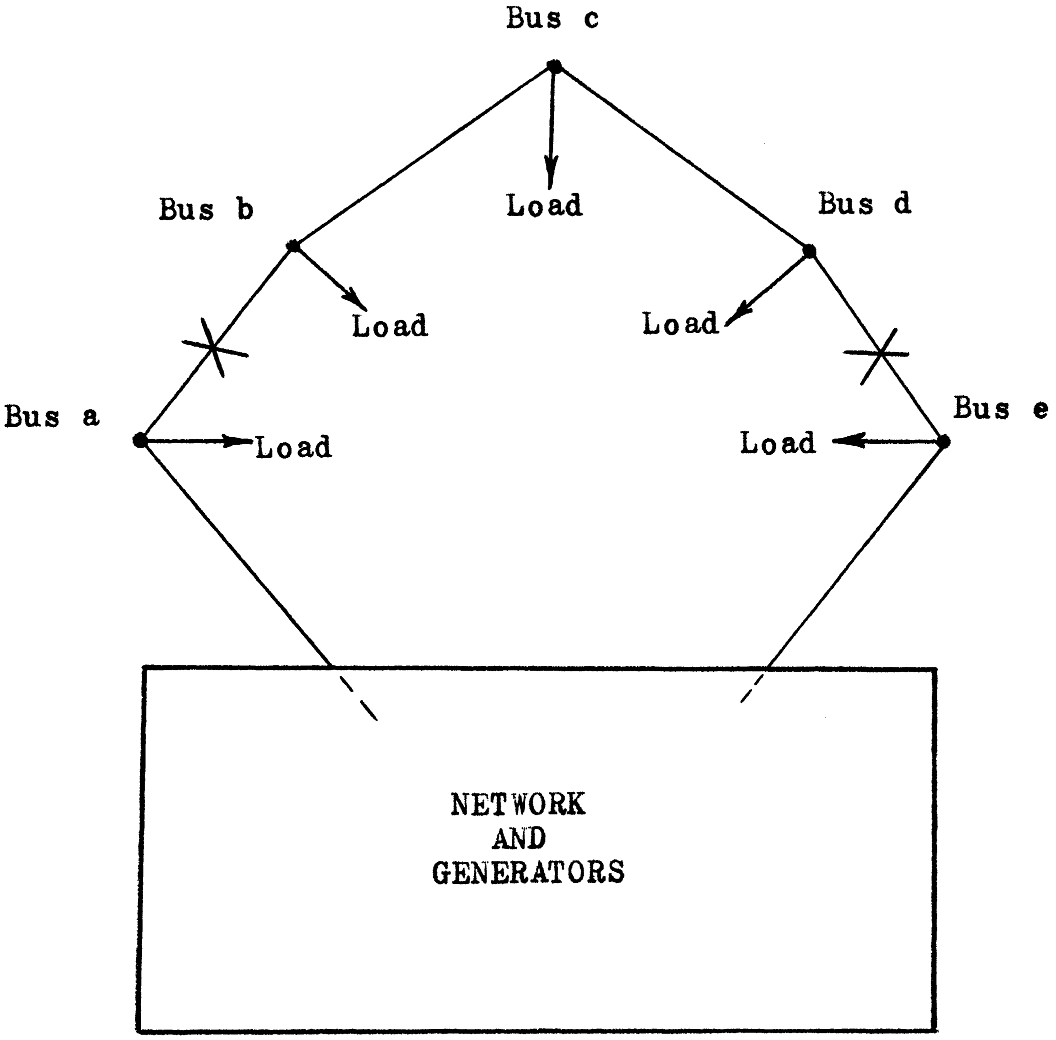
Some discussion on the motivation behind the particular


Start
Set C
$P_i \leftarrow P_1$
$\overline{X} \leftarrow \overline{X}^i_{min}$
$\overline{XF} \leftarrow \overline{X}$
$P_i \leftarrow P_{i+1}$
$f(A_i, \overline{X}, \overline{XF}, \overline{B}) > C$
no
yes
Rule 2 & 3
yes
$i < 2^M - 1$
no
Stop
$\overline{V}(\overline{X}^i_{max}, \overline{B}, s_j) > \overline{V}^{s_j}_{min}$
$\forall s_j$
yes
no
Rule 1
no
$f(A_i, \overline{X}, \overline{XF}, \overline{B}) > C$
yes
$\overline{V}(\overline{X}, \overline{B}, s_j) > \overline{V}^{s_j}_{min}$
$\forall s_j$
$\overline{XF} \leftarrow \overline{X}$
yes
no
no
$\overline{X} \leftarrow \overline{X}+1$
Overflow
yes
$\overline{V}(\overline{X}, \overline{B}, s_o) > \overline{V}_{max}$
SKIP OPERATION
no
yes
Rule 4
Rule 6
Rule 5
FEASIBLE SOLUTION
Set C

IMPLICIT ENUMERATION ALGORITHM
FOR FIXED CAPACITOR ALLOCATION

FIGURE 12

enumeration sequence used is now in order. Two considerations provide this motivation. The overall objective of optimal capacitor allocation is, of course, to obtain the least expensive allocation(s) which will satisfy the voltage requirements of the system. Particularly in cases where equipment outages must be taken into consideration, one possibility for the least costly solution would be capacitor banks applied at certain "critical" buses which are influential under more than one of the outage conditions. Figure 13 shows a hypothetical network in which such a "critical" bus might be found. Suppose that two line outages are to be considered—each indicated by an X on the diagram. In one case power would be fed from the network through a, b, and c to d. In the other case, flow would be through e, d and c to b. In the former, a low voltage bus might result at d, in the latter at b. Assume that this is the case. Now it might happen that the first outage could be satisfied by either 2 units at d or 3 at c and the second outage could be satisfied by 2 at b or 3 at c. Thus two solutions to the problem might be 2 at d and 2 at b or 3 at c (depending, of course, upon whether the high voltage constraints are satisfied or not). In this case the solution requiring that least number of units involved only one bus. It, in a sense, was "critical" in that it provided the least costly solution to two outage conditions. Other situations similar to this are conceivable. Thus the strategy taken in the enumeration sequence is to search for the "critical" buses first by considering the allocations with the smallest application subsets first.



Hypothetical Network Configuration
Containing a "Critical" Bus

FIGURE 13

Notice that until critical buses are located, the algorithm will remain in the loop involving Rule 1 in Figure 12.

Another consideration influencing the choice of this particular enumeration sequence involves the number of vectors in each product space P_i. As the sequence proceeds, the number of vectors in each product space increases. Their number is not monotone nondecreasing, but in general the latter product spaces will have more vectors than the earlier ones. Of course, the number of operations involved in enumerating all the vectors in these spaces increases with the number of vectors to be enumerated. The order used in the product space sequence delays consideration of those containing large numbers of vectors as long as possible. This is done in the hope that a feasible solution will be obtained in the earlier spaces that may enable the rejection of the larger spaces later in the sequence on the basis of cost considerations (Rules 2 and 3). If "critical" buses should prove to be important in a particular problem, it may turn out that, after the first few solutions have been obtained, most of the allocations remaining to be examined can be rejected.

Once a point in the enumeration sequence has been reached where all remaining allocations are of greater cost than some known feasible solution (or cost C'), then the algorithm remains in the loop involving Rule 2 (or 3) until the product space sequence has been completed. The operations involved in this loop do not involve much computation and are quickly executed. Note that, in general, it is not possible to terminate execution once this loop has been entered or even re-entered several times. This is particularly the case when cost is a function of the application subset involved.[†] The reason is that the lower cost bounds for the P_i may not be monotone nondecreasing as the sequence proceeds. Extra logic, to determine when termination can occur, may or may not be advantageous. With certain types of cost functions, logic to prematurely terminate the product space sequence can easily be accomplished. Such logic is incorporated in the program to be discussed in Chapter 8.

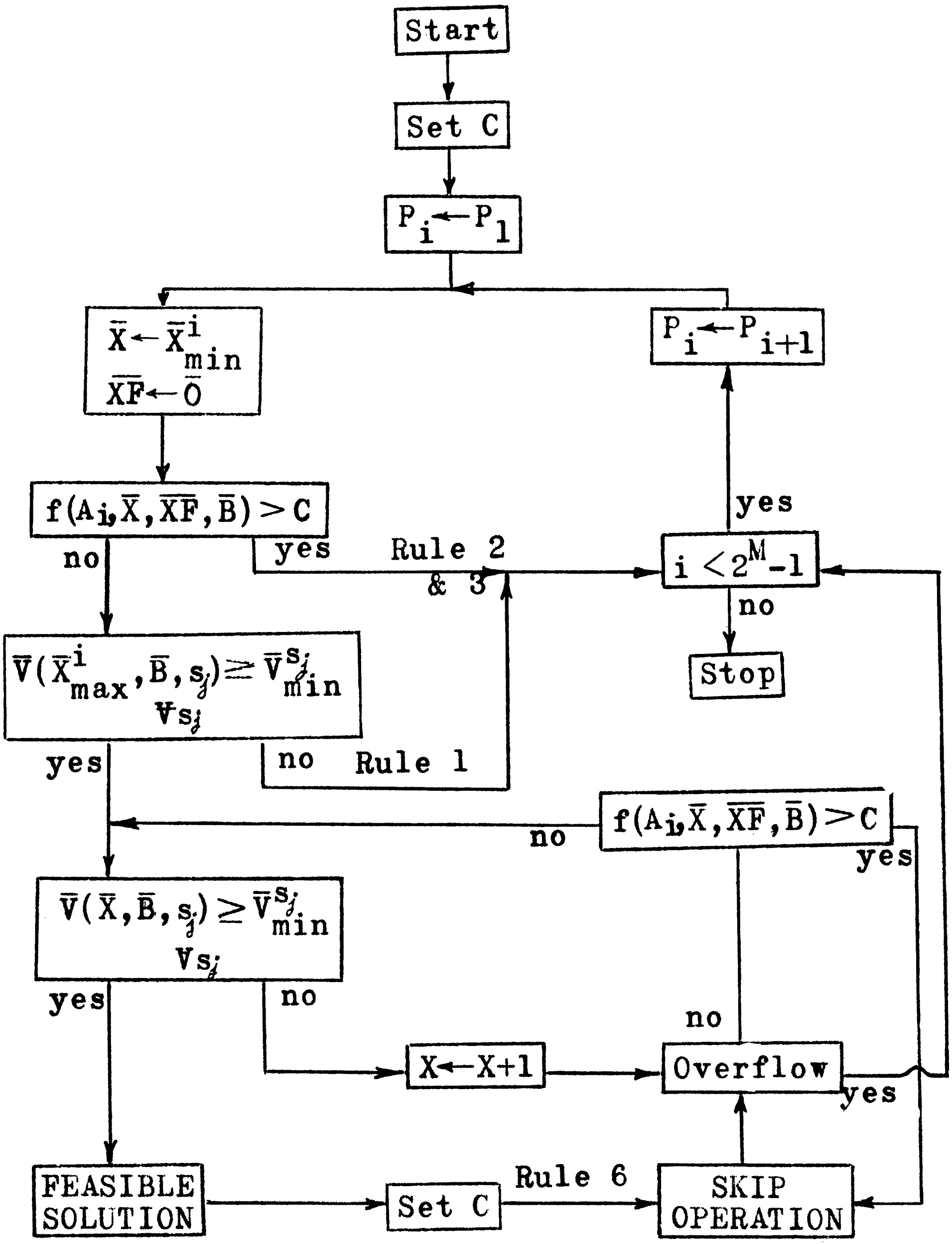
6.6 Modification for Switched[††] Capacitors

The algorithm described in the last section can find the optimum allocation of fixed static capacitors for a given set of application buses. It will sometimes be of interest to determine the optimal allocation of switched static capacitors. In fact, in some utilities, no fixed units at all are applied on the higher voltage levels. Consumers Power, for example, uses switched banks exclusively on networks of 46KV and up.

[†] Cost functions of this type will be discussed in Section 8. 4.

[††] In this section it will be assumed that switched banks are removed and inserted as a whole during switching.

An important constraint in the allocation procedure for fixed capacitors is the maximum voltage bound. In the case of switched capacitors, this constraint is not as critical, as the capacitors can be removed from the network when they cause voltages above a certain level. It is known that the voltages accompanying state s_0 can always be kept below the maximum $\bar{V}_{max}$ simply by removing all of the switched capacitors from the network. (This assumes, of course, that there are no overvoltages in state s_0 to start with.) Therefore the constraint involving $\bar{V}(\bar{X}, \bar{B}, s_0)$ need not be considered. One possibility, however, is that for one of the states, s_j, so many capacitors are added at a bus, in bringing up the voltage of a neighboring bus, that it itself exceeds the high voltage limit. To prevent this, constraints of the form $\bar{V}(\bar{X}, \bar{B}, s_j) \leq \bar{V}_{max}$ would have to be incorporated in the algorithm. This would, of course, involve additional computation. The probability of these constraints being violated in practical situations is not very large. The low voltage buses are usually fairly far removed from those of higher voltage so that banks at the higher voltage buses alone will not aid sufficiently enough for the low voltage constraint to be satisfied. Allocations which involve capacitors at both high and low voltage buses could satisfy the low and violate the high voltage bound, but such allocations would be rejected by the algorithm because a less expensive solution involving only the low voltage buses would be generated first in the enumeration sequence. (Note that rejection might not occur if the


Start
Set C
$P_i \leftarrow P_1$
$\bar{X} \leftarrow \bar{X}^i_{min}$
$\bar{XF} \leftarrow \bar{0}$
$P_i \leftarrow P_{i+1}$
$f(A_i, \bar{X}, \bar{XF}, \bar{B}) > C$
no
yes
Rule 2 & 3
yes
$i < 2^M - 1$
no
Stop
$\bar{V}(\bar{X}^i_{max}, \bar{B}, s_j) \geq \bar{V}^{s_j}_{min}$ $\forall s_j$
yes
no
Rule 1
$f(A_i, \bar{X}, \bar{XF}, \bar{B}) > C$
no
yes
$\bar{V}(\bar{X}, \bar{B}, s_j) \geq \bar{V}^{s_j}_{min}$ $\forall s_j$
yes
no
no
$X \leftarrow X+1$
Overflow
yes
FEASIBLE SOLUTION
Set C
Rule 6
SKIP OPERATION

IMPLICIT ENUMERATION ALGORITHM FOR SWITCHED CAPACITOR ALLOCATION

FIGURE 14

user specified that all solutions less than some amount were to be found and this amount fell above the cost of the undesirable allocation.) As an added precaution, however, a slight modification can be made in the calculation of the $XM_0(\alpha_I)$. In some cases it may be that the difference between the present voltage at a bus and the maximum voltage allowed at that bus is less than the maximum voltage rise used in the calculation of $XM_0(\alpha_I)$. Clearly, the smaller value of the two should be used in the calculation. This would at least prevent any installation at a single bus from exceeding the high voltage bound.

It shall be assumed here that any switched allocation that satisfies the low voltage constraints is a feasible solution.[†] The same algorithm as that used for fixed capacitors can be applied by bypassing the test involving the high voltage constraints. Of course, the cost function must now include the cost of circuit breakers and any associated control systems. A flow diagram of the resulting algorithm is shown in Figure 14.

6.7 Modification for Simultaneous Allocation of Fixed and Switched Capacitors

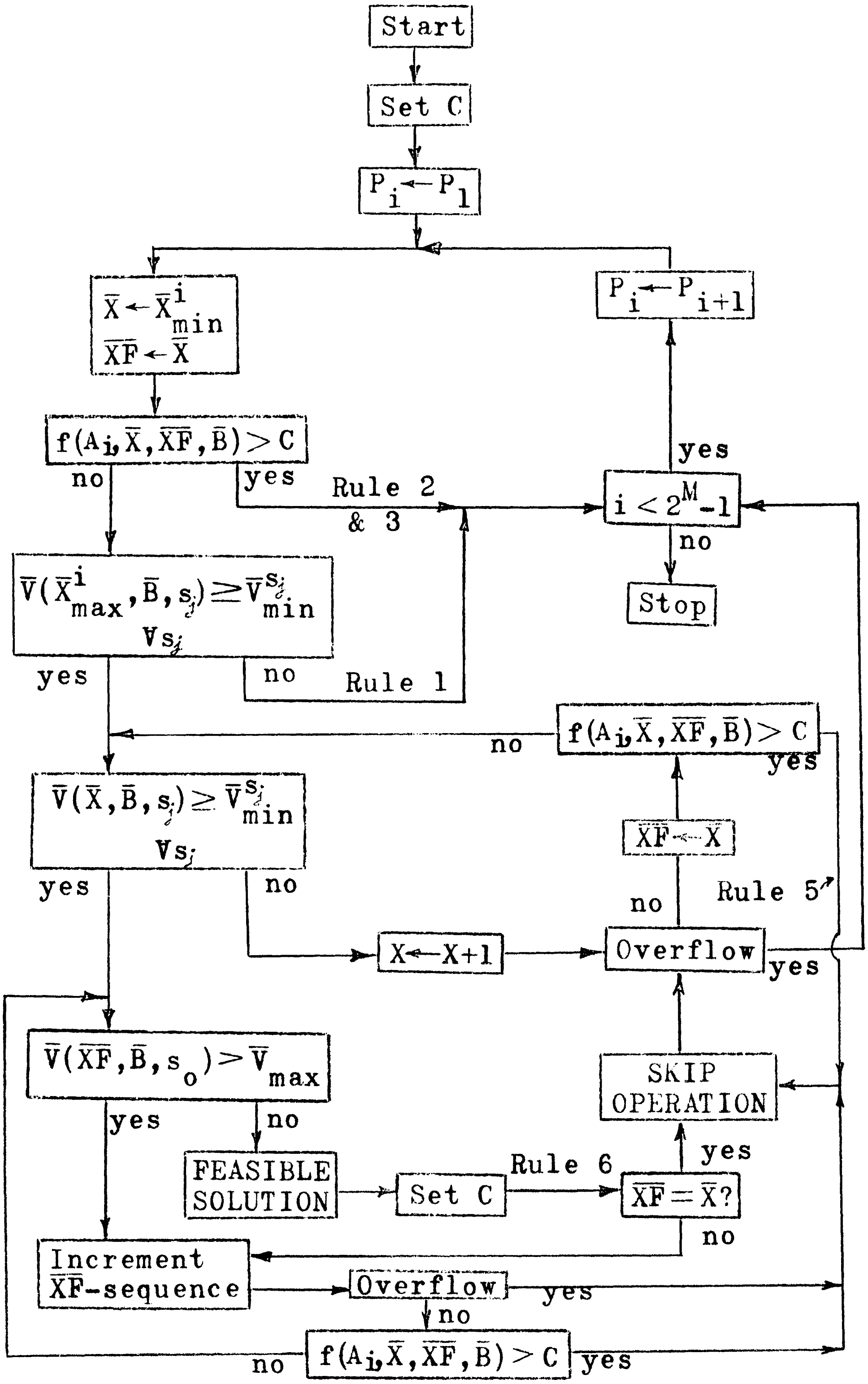
In view of the simplicity and lower cost of fixed capacitor banks

----

[†] Even if this did turn out not to be the case, the erroneous solutions could easily be rejected after the algorithm had run.

(no circuit breakers or controls being required for switching), the engineer may prefer to use them wherever possible and only use switched banks when absolutely necessary. The implicit enumeration algorithm can be modified to perform this kind of allocation too.

Figure 15 shows a flow diagram of the modified algorithm. It is almost identical to the one for fixed capacitor allocation except for the operations following the violation of a high voltage constraint. The algorithm first attempts to find a fixed capacitor allocation that will satisfy the low voltage constraints. Once such an allocation has been obtained, it is examined to see if it also satisfies the high voltage constraint. If it does, a fixed capacitor solution has been obtained. If it doesn't, the algorithm performs a search to determine which of the banks must be switched in order to satisfy this constraint. This search appears in the form of a sequence of $\overline{XF}$-vectors each derived from the existing $\bar{X}$-vector. Each $\overline{XF}$ is identical to $\bar{X}$ except for the elements corresponding to switched banks—which are zero. First, all the $\overline{XF}$ are generated which can be obtained from $\bar{X}$ by setting only one element to zero. Then those are generated which are obtained by setting two elements to zero ... then three elements, etc. Progression in this sequence is indicated by the box "Increment $\overline{XF}$" in Figure 15. This is equivalent to generating, in increasing order of their cardinality, the switched subsets of the allocation subset corresponding to the original fixed allocation. This sequence


Start
Set C
$P_i \leftarrow P_1$
$\bar{X} \leftarrow \bar{X}^i_{min}$
$\overline{XF} \leftarrow \bar{X}$
$P_i \leftarrow P_{i+1}$
$f(A_i, \bar{X}, \overline{XF}, \bar{B}) > C$
no
yes
Rule 2 & 3
yes
$i < 2^M - 1$
no
Stop
$\bar{V}(\bar{X}^i_{max}, \bar{B}, s_j) \geq \bar{V}^{s_j}_{min}$
$\forall s_j$
yes
no
Rule 1
$f(A_i, \bar{X}, \overline{XF}, \bar{B}) > C$
no
yes
$\bar{V}(\bar{X}, \bar{B}, s_j) \geq \bar{V}^{s_j}_{min}$
$\forall s_j$
$\overline{XF} \leftarrow \bar{X}$
yes
no
no
Rule 5'
$X \leftarrow X+1$
Overflow
yes
$\bar{V}(\overline{XF}, \bar{B}, s_o) > \bar{V}_{max}$
SKIP OPERATION
yes
no
yes
FEASIBLE SOLUTION
Set C
Rule 6
$\overline{XF} = \bar{X}?$
no
Increment $\overline{XF}$-sequence
Overflow
yes
no
$f(A_i, \bar{X}, \overline{XF}, \bar{B}) > C$
no
yes

IMPLICIT ENUMERATION ALGORITHM FOR FIXED AND SWITCHED CAPACITOR ALLOCATION

FIGURE 15

is purposely ordered in such a fashion that the least expensive allocations occur first. If the cost of converting from fixed to switched banks is the same regardless of the bus involved, a sequence of allocations results which is monotone nondecreasing in cost. This permits the use of Rule 2 or Rule 3 to terminate the search for appropriate switched units. Once all of the units have been made switched (Overflow) or the cost bound has been violated, return is made to the $\bar{X}$-vector sequence through the SKIP OPERATION.[†] This is justified because all of the allocations in the sub-sequence skipped would only have additional capacitor units at each bus of the allocation just considered. No additional useful information could be gained from their consideration.

The flow diagram should be pretty well self-explanatory in view of the past discussion. However some explanation may be needed for the box "$\overline{XF} = \bar{X}$?" just after the lower "Set C" operation. This operation is intended to determine whether the feasible solution obtained was a fixed allocation or not. If it wasn't, search is directed to finding other possible combinations of switched banks in the same vector $\bar{X}$ which involve no additional cost for switch-gear and controls.

[†] In the general case where the cost of switch-gear is a function of bus location, it will be necessary to wait for an "overflow" before proceeding to the SKIP OPERATION. In this case, the cost of the $\overline{XF}$-sequence may not be monotone nondecreasing.

It was mentioned in Section 3. 4 that limits are placed on the maximum number of capacitor units in a bank to limit voltage transients if it is switched or for reliability purposes if it is fixed. It should be noted that in the solution procedure used here, there is no way of knowing ahead of time whether a capacitor bank will be fixed or switched in a given allocation. This means that the $XM_0(\alpha_I)$ must be calculated for the type of bank with the smallest allowable voltage rise. The smallest allowable rise will usually be for switched banks.

6. 8 Flexibility of the Algorithms

Before concluding the chapter, it is important to note that the implicit enumeration algorithms depend for correct operation only on the monotone nondecreasing properties of voltage and cost with capacitor additions. Therefore any means of determining voltage levels will suffice. If so desired, a load flow could be run to determine voltage levels for each capacitor allocation. As faster load flow analysis programs are developed this may be the best approach. In the computer program to be discussed in a later chapter, the linear model discussed in Section 5. 3 is used. This was done solely for the purpose of speeding up computation. However, this does result in some loss in accuracy. More will be said about this later.

Chapter 7

AUTOMATED CAPACITOR ALLOCATION

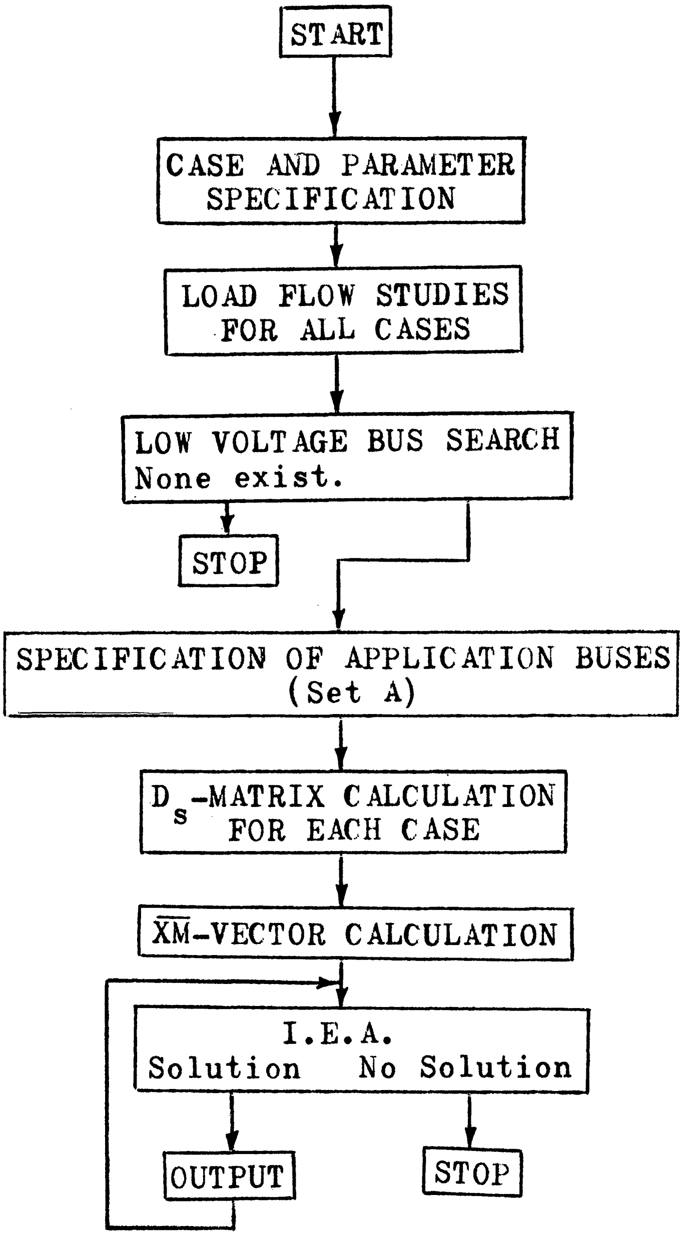
This chapter outlines a rudimentary capacitor allocation program and the data manipulation that it requires. Section 7. 1 begins with a general flow chart for the program and discusses the initial data operations. Each of the following sections handles a different aspect of the data preparation required for the Implicit Enumeration Algorithm. In some cases more than one approach to the data preparation is possible and the different possibilities are discussed.

7. 1 Input Data and Initial Operations

All that is necessary to form a rudimentary capacitor allocation program are the implicit enumeration algorithms just described and some means of supplying the input data they require. The flow diagram in Figure 16 shows the operations which must be performed. The program user must specify the network loads, generation and outage cases to be studied along with parameters such as capacitor unit sizes, allowable voltage range, costs and mode[†] of program operation. Load flow studies can then be performed with this data to establish voltage levels for each outage case, s_j, without additional capacitors installed. If any fixed capacitor units are to be

----

[†] The "mode" refers to whether fixed, switched, or fixed and switched capacitor units are to be applied.


START
CASE AND PARAMETER SPECIFICATION
LOAD FLOW STUDIES FOR ALL CASES
LOW VOLTAGE BUS SEARCH
None exist.
STOP
SPECIFICATION OF APPLICATION BUSES (Set A)
D_s-MATRIX CALCULATION FOR EACH CASE
$\overline{XM}$-VECTOR CALCULATION
I.E.A.
Solution
No Solution
OUTPUT
STOP

RUDIMENTARY ALLOCATION PROGRAM

FIGURE 16

applied, a load flow study must also establish the maximum voltage levels anticipated in state s_0. A search is then made to locate all buses with voltage below the minimum allowable value. If no low voltage buses exist, there is no need to continue. Execution terminates.

7.2 Specification of the Application Set

Specification of the application set, A, can be performed in various ways. The low voltage buses should certainly be among those included, but these alone may not result in the best solutions. The problem of specifying only the pertinent additional buses does not seem to have a solution that can be implemented with a reasonable amount of logic. However, there are several possible approaches which should prove adequate for practical purposes. First, all buses which can have capacitors applied at them could be included in A. This approach, while being "brute force," has the advantage of being thorough. One is assured that no good possibilities will be missed. But since the operations required for the algorithm to obtain a solution increase with the cardinality of A, this thoroughness is purchased at the cost of additional computing time.

Another possibility is to use human pattern recognition abilities to select buses for A. The designer can select those buses which his experience tells him are the most appropriate. Any uncertainties on his part can always be overcome with the inclusion of additional buses.

There are two ways in which this program-human interaction could be accomplished. For "batch" operations the program could be broken into two parts at the point just after the low voltage buses have been determined. The first part could write into a temporary file all arrays and parameters needed in the second. For example, if further load flows are to be run in the second part, the arrays containing Y_{BUS} (or Z_{BUS}) should be saved. In addition, any parameters such as mode of operation, costs of capacitor units and controls, etc. could all be saved in this temporary file rather than resubmitted. After the additional application buses have been determined, they can be submitted, along with the material in the temporary file, to the second part of the program. Alternatively, the program could be run in conversational mode, with the engineer seated at a remote terminal. After the low voltage buses have been determined, the engineer is notified. He then enters the numbers of any additional buses he wishes considered.

There is yet a third possible approach. Each bus on the system is influenced to a certain extent by capacitor additions at other buses. Only the buses with a significant amount of influence on the low voltage buses will aid in finding an economical solution. If a quantity could be assigned to this notion of influence, a minimum significant value of influence might be established on the basis of experience with the allocation program. Then only those buses with more than this amount of influence on the low voltage buses need be included in the application set, A.

The following definition assigns a quantity to this notion of influence.

Definition:

The influence of bus β on bus α is the voltage magnitude change at bus α resulting from the addition of a single capacitor unit at bus β. It shall be denoted by $\Delta|E_\alpha|_\beta$. (Note: Influence is always defined relative to a particular capacitor unit size.)

Ideally, the method for calculating influence should not be too computationally involved. If load flow analysis was used for this purpose, one load flow would have to be run with one unit of capacitance at each bus whose influence on another bus was to be determined. To determine set A, this would mean one load flow for each bus on the system—a sizable undertaking. Another possibility is to use the equation $\overline{\Delta E} = Z_{BUS}\overline{\Delta I}$. But, as will be seen in the next section, Z_{BUS} may not be readily available. Happily, a simpler method can be derived. Its discussion will be in better context after the topic of the next section has been discussed.

7.3 The Voltage Change Model

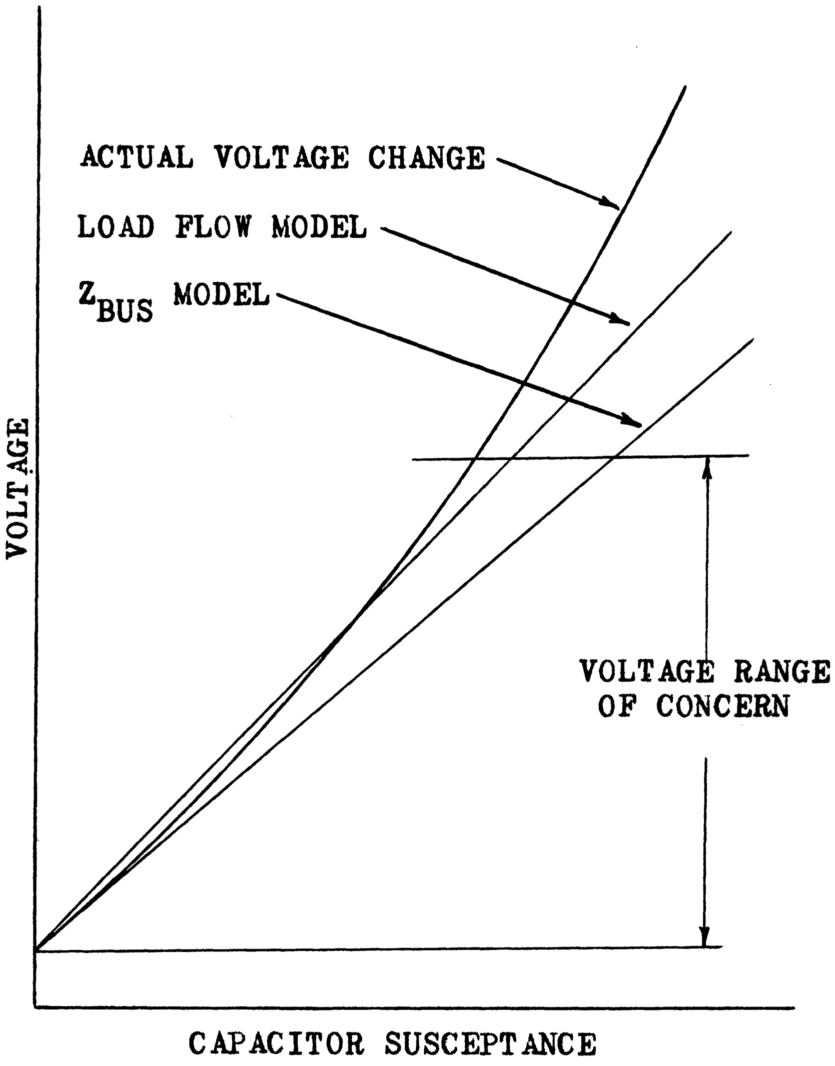
As mentioned in the last chapter, any reasonably accurate model of the voltage changes with capacitor additions will work satisfactorily as far as the algorithm is concerned. However, in the interest of obtaining some computational savings, the linear model discussed in Section 5.3 bears consideration. This involves the use of a matrix, D_{s_j}, for each system state, s_j, under consideration.

It would appear that the D_s-matrix calculations could be accomplished in more than one way. One method could make use of the equation $\overline{\Delta E} = Z_{BUS}\overline{\Delta I}$, or $\overline{\Delta E} = X_{BUS}\overline{\Delta I}$. The latter is essentially what is done by Maliszewski [24]. These approaches have two disadvantages. They require the calculation of Z_{BUS}[†] or X_{BUS} and they are not very accurate. Maliszewski solved the accuracy problem by first solving his linear programming problem, and then correcting the X_{BUS} model through the use of differences between the linear voltage prediction and the actual load flow results for that solution. The linear programming was then repeated. This process was continued until the error became less than a preselected tolerance.

[†] The system model already formed for the load flow program is Y_{BUS} in most cases. This is because the storage requirements are less for Y_{BUS} than for Z_{BUS}. Therefore Z_{BUS} or X_{BUS} would have to be calculated especially for the model.

An alternate method would be to use a load flow analysis to solve for the voltage changes resulting from the addition of a unit of capacitance at a given application bus. This would be repeated for each bus in set A. Since a small addition of capacitance will usually perturb the existing system voltages relatively little, only a few iterations should be required to converge to a solution from these existing voltages. Figure 17 compares the accuracy of the Z_{BUS} and load flow derived models. The curve shown is an exaggerated view of the voltage change at a bus as a function of capacitor susceptance somewhere in the system. The Z_{BUS} method gives a model based on a straight line drawn tangent to the curve at the initial voltage level. The error in the resulting voltage prediction increases as more capacitance is added. This error can be quite sizable over the voltage changes to be considered. But a model based on a straight line through two points on the curve can have less absolute error over the voltage range to be considered. The mean absolute error will depend upon the location of the second data point.

Both of these methods involve iteration at one time or another to improve the accuracy of the results. In Maliszewski's approach only the optimum solution is sought and iteration on the linear programming solution is feasible. In the approach presented here, many solutions may be sought and such iteration on each is impractical. The linear model must be as accurate as possible before the search for solutions begins. Therefore the load flow method of calculating



LINEAR MODEL COMPARISON

FIGURE 17

the D_s-matrices will be used in the program described in the next chapter.

7. 4 The Calculation of "Influence"

In Section 7. 2 "influence" was defined and reference was made to its use in selecting the application buses to be included in set A. This section discusses the calculation of influence.

One possible way of obtaining the approximate influence of one bus on another would be to use the equation $\overline{\Delta E} = Z_{BUS}\overline{\Delta I}$. But, unless Z_{BUS} is to be used in either the load flow or D_s-matrix calculations it would have to be calculated especially for this purpose. It would be nice if this additional calculation could be avoided. Fortunately it can. An alternative method will now be derived.

Consider the equation $\overline{\Delta E} = Z_{BUS}\overline{\Delta I}$. A capacitor addition at bus K can be regarded as the addition of a load current at bus K approximately equal to $- E_K^o(jB_c)$.[†] Here, B_c is the capacitor susceptance and E_K^o is the voltage at bus K before the capacitor addition. If all other load currents are assumed unaffected by this addition, $\overline{\Delta I}$ will consist of zeros except for ΔI_K which will be equal to $- jE_K^oB_c$. Therefore the change in voltage at bus J due to a capacitor addition at bus K is given approximately by $\Delta E_J = - jZ_{JK}B_cE_K^o$. In most cases

----

[†] See footnote on page 43.

$X_{JK} >> R_{JK}$ where $Z_{JK} = R_{JK} + jX_{JK}$. This means that $\Delta E_J \cong X_{JK} B_c E_K^o = \alpha E_K^o$ where α is a scalar. Now, if E_K^o does not differ appreciably in phase from E_J^o, it can be assumed that the change in voltage magnitude at bus J is approximately equal to the magnitude of the complex voltage change.[†] This means that $\Delta|E_J| \cong |\Delta E_J| \cong |-jZ_{JK} B_c E_K^o|$.

Let $\Delta|E_J|_K$ be the change in voltage magnitude at bus J resulting from a unit capacitor addition at bus K. Suppose the effect on system voltage magnitudes of a single capacitor addition at bus J is known. From the above reasoning it follows that

$$\Delta|E_1|_J \cong |-jZ_{1J} B_c E_J^o|$$

$$\Delta|E_2|_J \cong |-jZ_{2J} B_c E_J^o|$$

$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots$$

$$\Delta|E_K|_J \cong |-jZ_{KJ} B_c E_J^o|$$

$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots$$

$$\Delta|E_N|_J \cong |-jZ_{NJ} B_c E_J^o|$$

[†] Usually there is less than a ten degree difference for buses in the neighborhood of each other.

Now suppose the influence of each of the other system buses on bus J is desired. Clearly

$$\Delta|E_J|_K \cong |-jZ_{JK}B_c E_K^o| = |-jZ_{KJ}B_c E_J^o \frac{E_K^o}{E_J^o}|$$

$$= |-jZ_{KJ}B_c E_J^o|\ |E_K^o|/|E_J^o|$$

$$\cong \Delta|E_K|_J\ |E_K^o|/|E_J^o|$$

The quantity $\Delta|E_K|_J$ corresponds to the K-th element in the D_s-matrix column corresponding to the application bus J. Therefore if the D_s-matrix columns corresponding to the low voltage buses are calculated, this equation can be used to determine from them the approximate influence on each of the low voltage buses of every other bus on the system.

7.5 Calculation of the XM-Vector

Once the D_s-matrices have been calculated, $\overline{XM}_0$ can be calculated from the data included in them. Each element $XM_0(\alpha_I)$ is given by

$$XM_0(\alpha_I) = INT \left[\frac{(\Delta E_{\alpha_I})_{MAX}}{\underset{\forall s_j}{\max} (\Delta E_{\alpha_I}^{s_j})} \right] - INST(\alpha_I)$$

where

$(\Delta E_{\alpha_I})_{MAX}$ is the largest voltage rise that a capacitor bank at bus α_I is allowed to create.

$\Delta E^{s_j}_{\alpha_I}$ is the incremental voltage change resulting at bus α_I when one unit of capacitance is added at bus α_I in system state, s_j. $(s_j = 0, 1, \ldots, L)$.

$\mathrm{INT}(\sigma)$ is a function which maps σ into the largest integer n, such that $x \leq \sigma$.

$\mathrm{INST}(\alpha_I)$ is the integer number of capacitor units already installed at bus α_I

This completes the data which must be submitted to an implicit enumeration algorithm. The remainder of the program consists of executing the algorithm and either verifying each solution it obtains with a load flow analysis or printing the approximate new voltage levels as obtained from the voltage change model.

Chapter 8

THE STATIC CAPACITOR ALLOCATION PROGRAM

This chapter discusses a computer program which uses the algorithms of Chapter 6 for static capacitor allocation. Section 8. 1 describes the program structure and its features. As a planning aid to those who may wish to use the program or a variation of it, the amount of array storage it requires is discussed in Section 8. 2. Then Section 8. 3 presents some computational experience on test systems. Section 8. 4 concludes the chapter with a discussion of some modifications which could possibly extend the program's usefulness and efficiency.

8. 1 Program Structure and Features

The main purpose of this research has been to develop a computer program which could automatically determine the least costly static capacitor allocations for solving voltage maintenance problems on power system networks. The theory underlying such a program has been presented in the preceding chapters. In this chapter, a program derived from this theory is discussed and its performance on test problems presented. The theory of the past chapters lends itself to many possible variations in approach. But so many modifications and variations are possible, that only suggestions of the various possibilities will be presented here. Each user will surely have his own ideas for modifications relevant to his own particular needs.

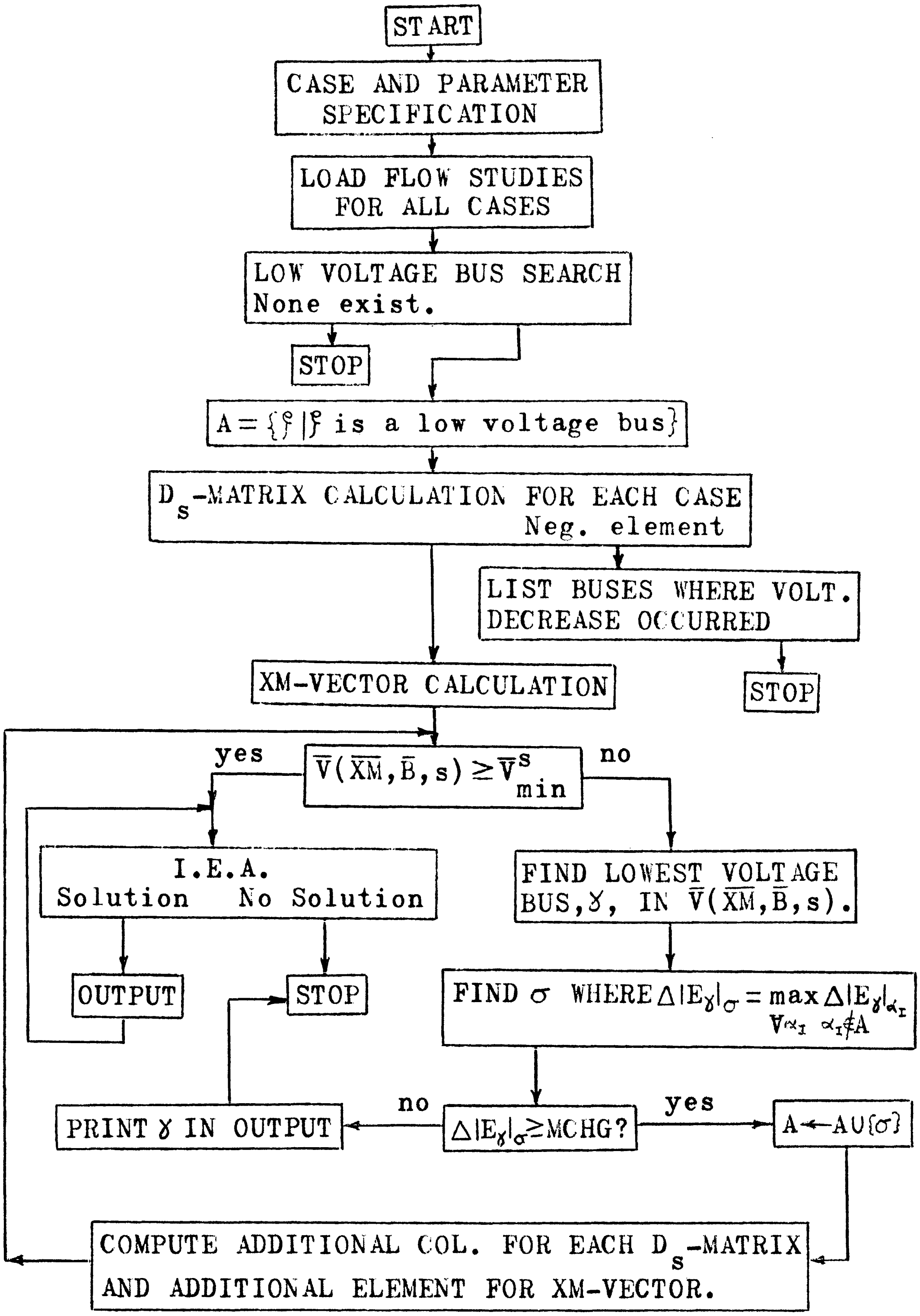
Therefore, the program presented incorporates primarily the basic method as described in the theory, with only a few embelishments. The program is intended primarily to test the feasibility of the discrete programming approach to capacitor allocation. But it is constructed in such a way that many of the variations in approach can be incorporated with little effort. FORTRAN IV coding is used so that it can be run on most computer systems without major rewriting.

The basic allocation program which has been developed is listed in the Appendix. For reference purposes it shall be referred to as the Static Capacitor Allocation Program or SCAP. Three different types of allocations are possible with SCAP—fixed banks only, switched banks only or both fixed and switched banks. The particular type desired is specified through an input parameter. In each of these, there is the capability of examining up to three different system "states" simultaneously. For fixed or fixed and switched bank allocation, one of these states must correspond to s_0—the state with the highest anticipated voltages. The other two can then represent anticipated low voltage conditions. For switched capacitor allocation, all three states can represent low voltage conditions. For switched capacitor allocation, all three states can represent low voltage conditions. The present program structure permits one line or transformer outage for each of the low voltage states. However, this can easily be extended to multiple outages if so desired. Each outage is specified by giving its terminal buses and circuit number. Also, two different load and generation schedules can be specified—one corresponding to light load conditions and the other to heavy load conditions. The effects of any capacitor banks already installed on the system can be included for each of these conditions. In this scheme existing switched banks can be removed from the network for the light load conditions and connected for the heavy load conditions. All specified existing banks are taken into account in the calculation of allocation costs and in the calculation of the $\bar{X}M_0$-vector which specifies the maximum number of additional units allowed at any bus. In some instances no further units can be applied at a bus due to the fact that its existing capacitor installation already provides the maximum allowable voltage rise. In the present coding all existing banks on the system are assumed to be switched and no additional cost for switch-gear and controls is assigned to any further unit additions to that same bank. The cost of switch gear and controls is included for any switched banks which would be installed in new locations.

All input data is submitted to the program on cards. This includes the specification of upper and lower voltage bounds (here assumed to be uniform across the system) and the unit size of capacitors to be installed (again assumed to be uniform). If a cost threshold is specified, all solutions with cost less than this amount are found by the program. If no such threshold is supplied, the program searches for the optimum solution by updating the cost threshold with each feasible solution obtained. This process has been discussed earlier. A bus influence threshold is also specified. Any two buses which have less than this amount of influence on each other are considered to have effectively no (zero) influence. Other data input includes that for generation, loads and the network itself. Card formats for all input data are given in the Appendix.

Figure 18 is a flow diagram of SCAP. After the initial data input, the exist. ıg voltages are determined for each of the system states to be considered. This is done with a load flow analysis subprogram. A search is then made for buses with less than the specified minimum acceptable voltage level. If none exist, there is no problem and execution terminates. If any low voltage buses are found they are included in application set A. Initially these are the only buses included in A.

SCAP makes use of a linear voltage change model derived from load flow studies. This was discussed in the last chapter. Two units of capacitance are applied to a bus in set A and the resulting voltage change vector, divided by two, gives the corresponding D_s-matrix column for the state in question. This is repeated for each bus in A and then the whole process is repeated for each of the other system states under consideration. Some reduction in the time required for each of the necessary load flows to converge to a solution is accomplished by suitable initialization of the bus voltages. Instead of initially setting each bus voltage to that of the slack bus, they are set to


START
CASE AND PARAMETER SPECIFICATION
LOAD FLOW STUDIES FOR ALL CASES
LOW VOLTAGE BUS SEARCH
None exist.
STOP
$A = \{\rho \mid \rho$ is a low voltage bus$\}$
D_s-MATRIX CALCULATION FOR EACH CASE
Neg. element
LIST BUSES WHERE VOLT. DECREASE OCCURRED
STOP
XM-VECTOR CALCULATION
yes
$\overline{V}(\overline{XM},\overline{B},s) \geq \overline{V}^s_{min}$
no
I.E.A.
Solution
No Solution
OUTPUT
STOP
FIND LOWEST VOLTAGE BUS, γ, IN $\overline{V}(\overline{XM},\overline{B},s)$.
FIND σ WHERE $\Delta|E_\gamma|_\sigma = \max_{\forall\alpha_I \; \alpha_I \notin A} \Delta|E_\gamma|_{\alpha_I}$
PRINT γ IN OUTPUT
no
$\Delta|E_\gamma|_\sigma \geq$ MCHG?
yes
$A \leftarrow A \cup \{\sigma\}$
COMPUTE ADDITIONAL COL. FOR EACH D_s-MATRIX AND ADDITIONAL ELEMENT FOR XM-VECTOR.

THE STATIC CAPACITOR ALLOCATION PROGRAM

FIGURE 18

the existing voltage for the system state under consideration.[†] As two units of capacitance perturb the system voltages relatively little, only a few iterations are required to converge to the new system voltages. This approach requires the storage of one complex voltage vector for each of the states being considered, but the resulting savings in computation may justify this in cases where there is not a shortage of computer storage.

As the D_S-matrix columns are calculated, any element which is smaller than the specified minimum bus influence, MCHG, is set to zero. In large power systems, buses lying at large distances from an applied capacitor will experience little or no raise in voltage. This means that many of the elements in each column will be zero. If the ratio of zero to nonzero elements is sufficiently high (say in the order of 5/2 or more) the information contained in the array can be compressed into less storage space by storing only the nonzero terms together with knowledge of their positions. This is accomplished with three smaller arrays. One stores the nonzero elements in column order and a second stores the row number in the D_S-matrix of the corresponding element in the first array. A third array stores the array entry number in the first (or second) array which corresponds to the last element in each D_S-matrix column stored. Figure 19 shows an example of this storage procedure. As forty or more buses may be

----

[†] Usually the slack bus voltage is the "best guess" for system voltages to start the iteration procedure. In this case a closer estimate is available from previous load flow studies.

$$\begin{array}{c|cccc}
\text{row} \diagdown \text{col} & 1 & 2 & 3 & 4 \\
\hline
1 & 0 & 0 & e & 0 \\
2 & 0 & 0 & f & 0 \\
3 & a & 0 & 0 & 0 \\
4 & 0 & 0 & 0 & h \\
5 & b & c & g & 0 \\
6 & 0 & 0 & 0 & i \\
7 & 0 & d & 0 & 0
\end{array}$$

Array Entry	N	R	P
1	a	3	
2	b	5	2
3	c	5	
4	d	7	4
5	e	1	
6	f	2	
7	g	5	7
8	h	4	
9	i	6	9

Array N contains the numerical values of the non-zero entries.

Array R contains the row position of the non-zero entries.

Array P contains the position of the end of each column in array N.

SPARSE MATRIX STORAGE

FIGURE 19

affected by a single capacitor addition, the technique has no advantage in systems of 100 buses or less. SCAP, although dimensioned for a 100-bus system, utilizes the three array "sparse matrix" storage technique.

If a voltage drop occurs at some bus when capacitors are added, this is an indication that something is amiss. Computation cannot be allowed to continue under such circumstances as the implicit enumeration algorithm depends for correct operation upon a monotone nondecreasing model for voltage change. The most probable cause of such behavior is incorrect specification of the bus influence threshold and the load flow convergence tolerance. As mentioned in Section 2.5, the convergence tolerance is a quantity which indicates to the load flow program when the voltages obtained are sufficiently accurate. A bus that is not significantly affected by a capacitor addition may differ in voltage from one load flow analysis to the next by any amount less than the convergence tolerance. The result may be that some bus voltages will appear to decrease with capacitor additions. If these apparent decreases in voltage are not rejected by the selection of an appropriately large bus influence threshold, MCHG, execution will terminate as shown in Figure 18. Other causes for a decrease in solution voltage may require a knowledge of the buses at which the decreases occurred in order to determine the cause. Therefore all such buses are listed before execution is terminated. Only in a rare instances will such decreases in voltage be due to actual system conditions and not some error in data specification or program operation.

Once the D_s-matrices have been successfully computed, the $\bar{X}M$-vector can be computed from the information contained in them. This computation is performed exactly as discussed in Section 7. 5.

The calculation of the $\bar{X}M_0$-vector completes the data required by the Implicit Enumeration Algorithm. But before proceeding to it, a check is made to determine if maximum capacitor additions at each of the buses in A can maintain system voltages above the minimum in all system states. If a state is encountered where $\bar{V}(\overline{XM}, \bar{B}, s_j) < \bar{V}^{s_j}_{min}$ then there is no reason to proceed to the algorithm. In this event, a search is made for additional buses which will rectify the situation when included in set A. Selection is made on the basis of bus influence on the lowest voltage bus remaining after the maximum capacitor additions. Influence is calculated by the method described in Section 7. 4. The search for additional buses is terminated when those not in A no longer have significant influence on the lowest voltage bus (i. e., if $\Delta|E_\gamma|_\sigma <$ MCHG where MCHG is the minimum significant influence). In such a case, no capacitor allocation can maintain system voltage. Knowledge of the lowest voltage bus remaining after capacitor additions can be helpful in determining the region of the network which must be aided by other types of equipment additions.

Therefore the number of this bus is printed out before execution is terminated.

Once a set A has been determined such that the condition $\overline{V}(\overline{XM}, \overline{B}, s_j) \geq \overline{V}^{s_j}_{min}$ is satisfied for all states, s_j, under consideration, then the implicit enumeration algorithm is called. Solutions obtained by the algorithm are listed on a printer as they occur. Two different types of output are possible—the one desired being specified by an input parameter. In one, complete load flow solutions are presented for each of the allocations found. Each solution is accompanied with indication of the location and size of the largest error occurring in the voltages predicted by the linear model. In the second type of output available, only the system voltages are listed. These are load flow derived voltages for the initial states without additional capacitors, but for the voltages resulting from each allocation they are only the approximate voltages given by the linear model. This output requires the least additional computation and is therefore the one which will probably be used most frequently.

This summarizes the salient features of the Static Capacitor Allocation Program. Further details are best obtained by reference to the program listing itself. This listing has been supplied with frequent comment statements to guide in its study. In addition, a glossary of variables used in the program has been included in the Appendix.

8.2 Array Storage Requirements

The storage requirements of many power system analysis programs are dominated by the large amount of array storage required for system data. The array storage requirements of SCAP will be briefly discussed in this section as a planning aid for the prospective user. General array requirements will be presented in terms of the following quantities:

N = number of system buses

M = number of elements in application set A

S = number of system states considered.

To provide some means of comparison, the requirements for a load flow program will be presented first. There are many different types of load flow programs and each will have its own unique requirements. The load flow program used as a subprogram in SCAP makes use of the Y_{BUS} matrix, which requires relatively little storage space. Total array storage requirements for this particular load flow program are roughly $60 \times N$ real words and $32 \times N$ integer words. As N becomes large, these requirements can become quite sizable.

In the section on solution methods for discrete programming problems it was mentioned that some algorithms require large amounts of computer storage just for the data required in the search operation. The implicit enumeration algorithms presented here have quite modest requirements in this respect. Integer array storage of dimension $9 \times M$ is all that is required for search data and control.

Storage of the linear model matrices, D_s, can make up a large portion of the array requirements of SCAP, particularly for smaller power systems. But, when stored using sparse matrix storage techniques, the array requirements for each matrix do not vary appreciably as the number of system buses in increased. This is because the number of nonzero elements in each column do not increase appreciably as this is done.[†] Computational experience with a 74-bus system showed an average of less than 40 nonzero elements per D_s-matrix column.[††] Using 40 as an upper limit, the array requirements for the linear models are $(40 \times M) \times S$ real words and $(40 \times M + M) \times S$ integer words.

A large amount of array storage needed in SCAP is required by miscellaneous data. For example, storage is required for the different generation and loading schedules of the low and high voltage system states. In addition, further storage is required for things like the voltages predicted by the voltage model for each system state and for the voltage constraints they are to be checked against. In SCAP, these and other arrays of this type amount to approximately $27 \times N$ real words of storage. It is

----

[†] See page 111.

[††] All entries with magnitude less than .001 per unit were set to zero.

possible that some of this requirement could be reduced, but usually only with an increase in computation time as some of the vectors would have to be recalculated each time they are used. Other, less basic, versions of SCAP will probably require more storage of this type.

It should be noted that the load flow program is not actually needed at the same time as the implicit enumeration algorithm. And only those portions of the program that are needed at one time need be kept in the main memory. All other portions can be stored temporarily in tape or disk files.

At present SCAP is dimensioned to handle systems of up to 100 buses and a maximum of three system states. Application set A can contain up to twenty buses. Overflows in either the application set array or the D_s-matrix arrays are accompanied by diagnostic error messages in the program output.

8.3 Computational Experience

The results of applying SCAP to several example problems are given in this section. Four of the example problems involve a small six-bus system. Their purpose is primarily to demonstrate the different types of allocation that can be performed. They also provide simple test situations for the user setting up his own version of the program. Since no general conclusions about the performance of SCAP can be drawn from such a small, hypothetical system, an actual 74 bus subtransmission network from the Consumers Power system is also considered. It provides data on the time required to obtain a solution and the accuracy of the linear model in predicting system voltages for a given allocation.

The cost function used in the example problems is completely hypothetical. The actual costs of capacitor units, switch gear and controls will depend upon the voltage level at which they are to be applied. The current interrupting ability required for the switch gear will also affect cost and this may be a function of the bank's location within the network. These factors can be taken into account in the cost function if so desired, but for the purpose of demonstrating the operation of the program some costs will be arbitrarily selected. More will be said later about the manner in which voltage and location dependent costs can be taken into account.

The cost function calculation is performed in a separate subprogram of SCAP called ECON. It is therefore easily changed without disturbing the remainder of the program. A listing of the subprogram used for the example problems is given in the Appendix. In each example the capacitor unit was assigned a cost of $12, 500. The cost of switch gear, controls, labor, etc. for a switched bank, was set at $10, 000 and for a fixed bank, labor and miscellaneous expenses were set at $3, 000. Each of the costs stated in the examples are derived from these figures. In all of the examples the following problem specifications are used:

maximum voltage level (V_{max}) = 1. 1 per unit

minimum voltage level ($V^{s_j}_{min}$) = . 92 per unit $\forall$ s_j

susceptance of capacitor units (B) = . 05 per unit

maximum voltage rise per bank ≈ . 045 per unit

minimum significant bus influence = . 001 per unit

Figure 20 is a one-line diagram of the six-bus system to be used.† It is based upon a test system described by Ward and Hale [37]. The network includes two transformers with "off nominal"

†A single phase representation of the three phase network is shown. It is assumed that each phase is identically loaded so the system is analyzed on a single phase basis.

FIGURE 20

SIX BUS TEST SYSTEM

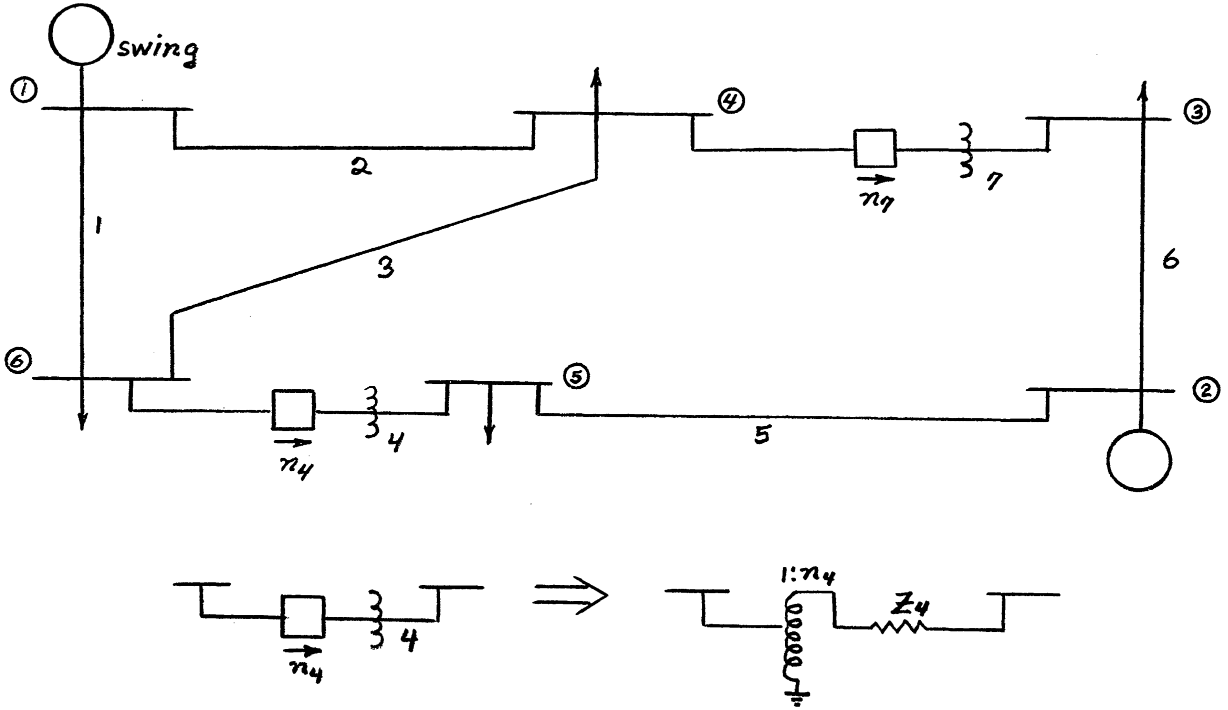


TABLE I

LINE AND TRANSFORMER DATA FOR SIX BUS TEST SYSTEM

Branch	R	X
1	.123	.518
2	.080	.370
3	.097	.407
4	.000	.300
5	.282	.640
6	.723	1.050
7	.000	.133

$n_4 = 1.0250$

$n_7 = 1.1000$

turns ratios.[†] The notation used for these is clarified by an equivalent circuit also shown in the figure. Table I lists the per unit impedances associated with the network.

In all of the system states to be considered the voltage magnitudes at buses 1 and 2 will remain fixed at 1. 05 and 1. 10 per unit, respectively. These buses have generators attached and are voltage regulated. Bus 1 is the slack bus and its voltage is assigned the reference phase of zero degrees. In the examples, each allocation is given in the order that it appeared in the solution process. A suffix F or S in the solution vectors indicates that the bank is fixed or switched, respectively. Specified terminal conditions (generation and loading) are given in per unit real and reactive power—a positive sign indicating generation. And finally, all voltages stated are derived from the load flow analysis program described in the Appendix.

[†] The turns ratios of the transformers in branches 4 and 7 are not identical. Therefore, it is not possible to choose base voltages for the two parts of the network separated by these transformers so that Equation (2. 4) is satisfied for both transformers simultaneously. This means that, even on a per unit system, each transformer cannot be represented as a single series impedance. However, their representation can be reduced to an equivalent π circuit (See [33], pp. 317-320). This is taken care of in the load flow program.

Example 1: Fixed Banks Allocated

Specified Terminal Conditions

State s_0: All lines in

Bus No. →	2	3	4	5	6
P	.20	-.20	-.10	-.10	-.45
Q	-	-.05	-.04	-.05	-.05

State s_1: All lines in

Bus No. →	2	3	4	5	6
P	.50	-.55	-.15	-.30	-.50
Q	-	-.13	-.05	-.18	-.05

State s_2: Same as line s_1 but line 3 removed

Existing Voltages

Bus No. →	3	4	5	6
$V_0^{s_0}$	1.0703	.9811	1.0099	.9771
$V_0^{s_1}$	.9576	.8922	.9017	.8929
$V_0^{s_2}$	.9649	.8975	.8975	.8877

Set A = $\{4, 5, 6\}$

$\overline{XM}_0 = (3, 2, 2)$

Solution 1 = (2F, 0, 2F)

Cost = ($56, 000)

Resulting Voltages

Bus No. →	3	4	5	6
V^{s_0}	1.0998	1.0114	1.0324	1.0101
V^{s_1}	.9882	.9230	.9252	.9261
V^{s_2}	.9932	.9262	.9208	.9233

Error in Linear Model

Bus No. →	3	4	5	6
ϵ_{s_0}	-.0003	-.0003	-.0002	-.0003
ϵ_{s_1}	-.0002	-.0002	-.0002	-.0002
ϵ_{s_2}	+.0009	+.0004	+.0007	-.0002

By running SCAP with the maximum possible cost threshold, it was found that the single solution found in this example is the only feasible solution to the problem.

Consider for a moment the effect of errors in the voltage prediction model. It should be mentioned that even voltages obtained from load flow analysis only approximate the voltages actually observed on the system. This is due to the fact that the loads specified for the analysis are themselves only approximate. It is impossible to know ahead of time exactly what the system loads will be—particularly when forecasts must be made for several years into the future. Of course, errors in the voltage prediction model will affect the allocations produced by SCAP. Suppose that the voltage rise predicted by the model for a unit capacitor addition is accurate to within ± x of the "actual" voltage rise. Then difficulties may arise when the voltage prediction is within ± x of one of the voltage bounds. In such situations SCAP may produce an allocation that is either pessimistic or optimistic. For example, if the predicted voltage should erroneously fall below a low voltage bound it will appear to the program that an additional capacitor unit is required. Similarly, with an erroneous prediction of overvoltage, it would appear that a fixed capacitor unit should be removed or the bank should be equipped with switch-gear. If the tolerance ± x is small in relation to the voltage rise created by a capacitor unit, then these effects have a tendency to produce a design with a certain margin of safety in cases where allocations may produce voltages near one or both of the voltage bounds. However, if the tolerance ± x is too large, "pessimistic" allocations containing more capacitor units or more switched banks than really necessary may result. It may also happen that the predicted voltages are erroneously greater than the low voltage bound or erroneously less than the high voltage bound. The "optimistic" allocations derived under such conditions are very undesirable. In them, the allocation will actually not maintain the desired voltage range. The chances of obtaining these are small if x is a small fraction of the voltage rise produced by the capacitor units.

In view of these considerations, it is interesting to note the size of the errors in the linear model as compared to load flow solutions. In this example the largest error occurred at bus 3. It was approximately .0007 per unit. The voltage rise at the same bus due to all of the installed capacitors was .0283 per unit. The error was therefore in the order of 3.2% of the total voltage rise. This should be satisfactorily accurate for system planning purposes.

Example 2: Switched Banks Allocated

Specified Terminal Conditions

State s_1: All lines in

Bus No. →	2	3	4	5	6
P	.50	-.55	-.15	-.30	-.50
Q	-	-.13	-.05	-.18	-.05

State s_2: Same as s_1, but line 3 removed

Existing Voltages

Bus No. →	3	4	5	6
$V_0^{s_1}$	.9577	.8922	.9018	.8931
$V_0^{s_2}$	.9650	.8976	.8953	.8878

Set A = {4, 5, 6}

$\overline{XM_0}$ = (3, 2, 2)

Solution 1 = (2S, 0, 2S)

Cost = $70,000

Resulting Voltages

Bus No. →	3	4	5	6
V^{s_1}	.9883	.9231	.9253	.9262
V^{s_2}	.9933	.9263	.9209	.9235

Errors in Linear Model

Bus No. →	3	4	5	6
ϵ_{s_1}	-.0002	-.0002	-.0002	-.0002
ϵ_{s_2}	+.0009	+.0004	+.0007	-.0002

In this example the optimum solution was the first feasible solution generated. When the program was re-run to obtain all of the feasible solutions to the problem, the following sequence resulted:

	Solution	Cost
1.	(2S, 0, 2S)	$70, 000
2.	(3S, 0, 2S)	$82, 000
3.	(2S, 1S, 2S)	$92, 000
4.	(2S, 2S, 1S)	$92, 000
5.	(3S, 1S, 2S)	$105, 000
6.	(3S, 2S, 1S)	$105, 000

This sequence points out one shortcoming of the algorithm when it is used to find suboptimal switched allocations. Only solutions 1 and 4 are of real interest. All the other solutions can obviously be derived from these by the addition of extra capacitor units at each bus. For example, solution 2 is simply solution 1 with an extra capacitor unit at bus 4. Some of the redundant solutions are rejected when the SKIP OPERATION is performed, but this affects only those in the vector partial ordering immediately following a solution. Those occurring later in the enumeration sequence are the ones that appear.

All redundant solutions could be rejected by comparing each successive feasible solution with those preceding it, but this would require additional storage for retaining the non-redundant solutions. Unfortunately, all algorithms of the search type used here will have this problem when used for an exhaustive search for all feasible solutions. They are most efficient when used to search for optimal solutions. However, the redundancy problem can be minimized by searching only for suboptimal solutions with cost just slightly greater than a known optimum.

Example 3: Fixed and Switched Banks Allocated

Specified Terminal Conditions

State s_0: All lines in

Bus No. →	2	3	4	5	6
P	.20	-.20	0.0	-.10	-.25
Q	-	-.05	0.0	-.01	-.05

State s_1: All lines in

Bus No. →	2	3	4	5	6
P	.50	-.55	-.15	-.30	-.50
Q	-	-.13	-.05	-.18	-.05

State s_2: Same as s_1 but line 3 removed.

Existing Voltages

Bus No. →	3	4	5	6
$V_0^{s_0}$	1.0925	1.0049	1.0390	1.0061
$V_0^{s_1}$	.9576	.8922	.9017	.8930
$V_0^{s_2}$	.9649	.8975	.8952	.8877

$$\text{Set A} = \{4, 5, 6\}$$

$$\overline{XM}_0 = (3, 2, 2)$$

Solution 1 = (2S, 0, 2S)

Cost = $70,000

Resulting Voltages

Bus No. →	3	4	5	6
V^{s_0}	1.0925	1.0049	1.0390	1.0061
V^{s_1}	.9882	.9230	.9252	.9261
V^{s_2}	.9932	.9262	.9208	.9234

Error in Linear Model

Bus No. →	3	4	5	6
ϵ_{s_0}	0.0	0.0	0.0	0.0
ϵ_{s_1}	-.0002	-.0002	-0.0002	-0.0002
ϵ_{s_2}	+.0009	+.0004	+.0007	-.0002

In this example, the optimal solution again turned out to be the first feasible solution found. Even though fixed and switched units were being allocated, no allocation with fixed units in it was possible. The linear model accuracy is similar to that obtained in Examples 1 and 2.

Example 4: Fixed and Switched Banks Allocated

All states in this example are identical to those in Example 3 except that a switched capacitor of .05 per unit susceptance is already present at bus 5. It is in the network for states s_1 and s_2 and disconnected for state s_0.

Existing Voltages

Bus No. →	3	4	5	6
$V_0^{s_0}$	1.0925	1.0049	1.0390	1.0061
$V_0^{s_1}$	.9610	.8958	.9181	.9014
$V_0^{s_2}$	.9644	.8973	.9153	.9008

Set A = {4, 5, 6}

$\overline{XM}_0$ = (3, 1, 2)

Solution 1 = (2S, 0, 2S)

Cost = $70,000

Resulting Voltages

Bus No. →	3	4	5	6
V^{s_0}	1.0925	1.0049	1.0390	1.0061
V^{s_1}	.9918	.9268	.9421	.9348
V^{s_2}	.9926	.9259	.9415	.9370

Solution 2 = (2S, 1S, 1F)

Cost = $66, 000

Resulting Voltages

Bus No. →	3	4	5	6
V^{s_0}	1. 0971	1. 0096	1. 0496	1. 0176
V^{s_1}	. 9902	. 9254	. 9508	. 9319
V^{s_2}	. 9925	. 9259	. 9489	. 9325

In this example, a suboptimal solution is generated in the process of obtaining the optimal solution. This suboptimal solution is identical to the optimal solution for Example 3. It is interesting to note how the existence of the switched unit already in the network is used to advantage in arriving at the most economical solution. The same number of capacitor units are added in both solutions, but the optimal solution avoids the addition of one set of switch gear and controls.

To obtain more realistic data on execution times and linear model accuracy, the Consumers Power 46KV Alma-Midland subtransmission network for 1968 was used in a further test. This network consists of 74 buses and 85 lines and has power supplied at eight buses. As no light load condition data was available only switched capacitor allocations were considered. The three network configurations or states included gave a total of 3 × 74 = 222 constraint equations to be checked for every allocation tested. The conditions were such that 15 low voltage buses existed in one or more of the three states. Only these low voltage buses were included in the application set. Capacitor banks already existed at most of these and two were found at which no further capacitance could be added without exceeding the voltage rise limitation of .045 p. u. Altogether there were 840 different possible allocations at the remaining buses

Under these conditions SCAP required about 14 seconds of IBM 360/67 CPU time to complete the three initial load flows establishing base case voltages. The three D_s-matrix calculations averaged about 44 seconds a piece. Once the Implicit Enumeration Algorithm had been called, it required 12 seconds to locate the first feasible solution. Then five other solutions were located, each requiring .113, .111, .161, .092 and .091 seconds, respectively. Total CPU time from data input to termination was 166 seconds.

Voltages derived from the linear models were checked against load flow solutions. The largest error at any of the low voltage buses was -.0026 per unit. In this case the error was approximately 12.2% of the total voltage rise at the same bus. This is worse than the maximum error found for the six bus system, but it is still not unreasonable. The largest error experienced for the 74-bus system was -.0037 per unit and it did not occur at any of the low voltage buses. In this case the error was exactly equal to the voltage rise—apparently because no voltage change had been predicted at that bus. This error is not surprising in view of the fact that all elements less than .001 in the D_s-matrices were set to zero. An error of several times this amount could easily accumulate as the number of capacitor units applied increased. Some of the error at the low voltage buses themselves is probably due to the neglect of the smaller elements in the D_s-matrices. These errors represent the largest ones encountered. For the most part, the other errors were less than .001 per unit. For system planning purposes, these errors should not be untolerable.

In none of the test problems solved on SCAP to date has there been any need to have other than the low voltage buses themselves in the application set, A. However, as found in the 74-bus test system, cases may arise in which no further capacitors will be allowed at some of the low voltage buses. In such cases, the bus addition logic incorporated in the program may prove useful.

8. 4 Modifications and Extensions to SCAP

In this section several of the possible modifications and extensions to SCAP will be discussed.

In the last section it was mentioned that the costs of capacitor banks will depend upon the voltage level at which they are applied and even their location within the network. The ability of a circuit breaker to interrupt short circuit current is one of the factors entering into its cost. The amount of short circuit current to be expected from a short in a capacitor bank will depend upon the bank's location within the network. As the possibility of such a short circuit must be planned for, the cost of switched capacitor banks must necessarily vary with their location. There is nothing in the underlying theory of SCAP which would prevent such considerations from being taken into account. The cost function will still be monotone increasing with each capacitor addition even though the amount of increase will depend upon the location and voltage level of the bank. In implementing such a nonuniform cost function, storage space for the cost data and some means of specifying the appropriate data to the program would have to be provided. If the short circuit currents at every bus on the system are known, a table of costs for breakers at each bus can be established.

It should be clear that, theoretically at least, there are many possible cost functions which could be utilized. In practice their usefulness will be determined primarily by their ease in implementation.

It should also be clear that there is no reason why the size of the capacitor units to be applied should be uniform across the whole system. With additional bookkeeping it could be arranged so that a different size would be placed at each bus on the network. Each column of the D_s-matrices would then correspond to the voltage rises for the unit size at a particular bus. The ability to handle nonuniform sizes will be important in situations where there is more than one voltage level on the network and there are different capacitor unit sizes for each level.

In the last section on computational experience, the CPU time required for some of the stages in setting up and solving the 74-bus system problem were given. These showed that the majority of the execution times were taken up in forming the linear model for the voltage changes and in performing the initial load flow studies. The actual algorithm time required was relatively insignificant. In recent years a considerable amount of work has been done in developing a method of load flow solution based on Newton's Method.[†] This work has been very successful and load flow programs are now becoming available that can solve large network problems in much less time than the Gauss-Seidel iterative method used in SCAP. SCAP can be relatively easily adapted to make use of such programs and any attempts to shorten SCAP execution time would do well to use this modification as a starting point. In fact, SCAP is constructed in such a way that any future developments in ways of solving for or approximating system voltage levels can be incorporated. No modifications to the underlying implicit enumeration algorithm will be necessary.

Computational experience with the 74-bus system also pointed out a possibly beneficial modification in the form of the data submitted to the implicit enumeration algorithm. Consider the following list of solutions as obtained in that test. Of the two numbers separated by a dash, the first is the bus number and the second is the number of

----

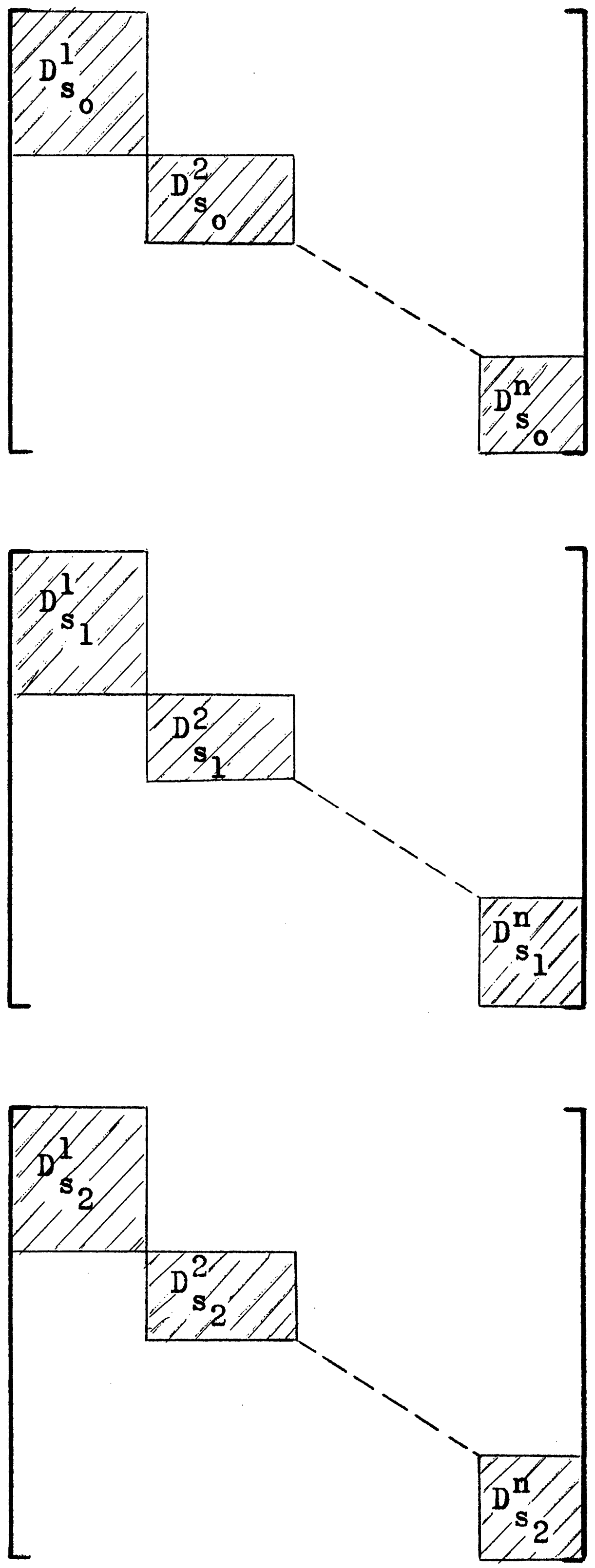
[†] See [33], pp. 270- 277 and [36].

switched units placed at that bus.

Solution 1:	13-1, 36-3, 50-1, 51-1
Solution 2:	13-1, 36-3, 50-1, 52-1
Solution 3:	13-1, 36-3, 51-1, 52-1
Solution 4:	13-1, 37-2, 50-1, 51-1
Solution 5:	13-1, 37-2, 50-1, 52-1
Solution 6:	13-1, 37-2, 51-1, 52-1

There is a recurring pattern in these solutions. One might suspect by looking at them that there are two (or possibly three) relatively independent regions of the network involved. One region has two possible solutions: (13-1, 36-3) and (13-1, 37-2). The other region has three possible solutions: (50-1, 51-1), (50-1, 52-1), and (51-1, 52-1). If there were some way in which the independence of these regions could be detected prior to application of the search algorithm, then the algorithm could be used to solve for the optimum in each region separately. This would remove the redundancy seen in the solutions above.

One means of detecting the independence of various regions of a power network would be to examine the D_s-matrices. If by permuting rows and columns it is possible to form matrices with diagonal structures as shown in Figure 21 (there being only zero elements outside the shaded areas), then the problem is obviously decomposable. If problem decomposition was incorporated into SCAP, the algorithm would have to be modified so that only the low voltage bounds in the



D_s-MATRIX FORM IN DECOMPOSABLE PROBLEM

FIGURE 21

network region being considered would be checked. At present, they are checked for the whole system. If the "pre-analysis" to determine regional independence can be accomplished rapidly, some improvement in execution time might result from problem decomposition.

Modifications, which may result in some decrease in execution time, are also possible within the implicit enumeration algorithm itself. Whether the resulting decreases will be significant or not, depends upon how much time the data preparation portions of the program require. The improvements may not give enough overall effect to be worth the effort. One obvious modification, which might shorten the algorithm search time, would be to perform all low voltage bound calculations and checks only on the low voltage buses themselves. In the present program, the approximate voltage at all system buses is calculated and checked against the low voltage bound every time the lower bound is checked. But, with the present data preparation times, any improvement resulting from changing these calculations would be relatively insignificant.

Finally, it should be mentioned that an implicit enumeration algorithm could even be used for the allocation of synchronous condensers. Their allocation is similar to the allocation of static units in the sense that they also come in discrete sizes or ratings. Unlike static units their output can be continuously varied, but their cost is determined by their rating in the same way as static units. There is no reason why an algorithm, similar to the one used for switched static units, could not be used to determine least costly synchronous condenser allocations. Of course, there will be additional details of bookkeeping to handle the different unit capacities available.

Chapter 9

CONCLUSIONS

In this chapter the contributions arising from the previously described work are summarized. In addition some suggestions are given for further research in capacitor allocation techniques.

9.1 Summary of Contributions

The preceding chapters have discussed a new formulation of the problem of static capacitor allocation for voltage maintenance. The problem is viewed as one of discrete optimization in which an allocation is chosen to minimize a cost function while at the same time satisfying system voltage constraints. By taking into account the discrete nature of the variables involved, solutions may be obtained which are less costly than could be found from continuous variable optimization techniques. The algorithms derived to solve the allocation problem have many assets—the primary ones being simplicity and flexibility. For example, they do not require excessive computer storage to direct the search for solutions. As power system analysis problems already tax some computer storage systems with the sheer volume of system data to be stored, this is an important attribute. Just as important is the variety in the types of allocations that the algorithms can consider. Their ability to handle fixed, switched, or fixed and switched static capacitor allocations as well as (with modification) allocations of synchronous condensers, makes them valuable tools on most voltage levels of the power system. Also of value is their ability to produce suboptimal solutions either as a by-product of the optimization process or by specific request. This is important particularly where design asthetics and reliability must be considered. These considerations are difficult to assign a dollar worth to so they must be taken into account after a set of economically good designs have been established. Suboptimal solutions are also important in long term planning where many design policies must be compared for several periods of time.

The Static Capacitor Allocation Program developed in the course of this research provides a framework for the many possible extensions and modifications of the method. Cost functions of various degrees of sophistication can be applied with minor modifications —all primarily of the "bookkeeping" type. The latest techniques for load flow analysis or voltage solution can also be incorporated as they become available. In addition, the program is adaptable for use on smaller computer systems by the use of "overlay" techniques in which only the portion of the program required at any one time is kept in main memory. This is possible because the load flow and algorithm subprograms are not needed simultaneously.

In short, the techniques and program that have been developed here provide a good foundation for capacitor allocation programs tailored to a variety of user's needs.

9.2 Suggestions for Further Research

The discrete programming approach to capacitor allocation has not yet been fully explored. One area of capacitor allocation for which the present algorithms do not apply is allocation for the purpose of minimizing losses on the power system. The function of capacitors in reducing system losses was discussed in Section 3.3. Again the problem is one of constrained optimization in which the capacitor additions appear as discrete variables. The object function to be minimized involves the system losses, although the costs of the capacitor units themselves might also be included. Constraints to be met could include system voltage levels and power factors. The formulation might also include system generation as a variable in which case it might be best described as a problem of mixed programming as both discrete and continuous variables occur. It is possible that an algorithm appropriate to the solution of this problem may already be described somewhere in the literature.

There is also a possible application for an algorithm, similar to the ones developed here for capacitor allocation, in the allocation of shunt reactors on transmission lines. At higher voltage levels the natural shunt capacitance to ground of transmission lines may cause overvoltages during light load conditions. This effect is sometimes corrected with the addition of synchronous condensers[†] or shunt reactors to ground. As the reactors also come in discrete sizes, the problem of their allocation is very similar to that for shunt static capacitors. An implicit enumeration algorithm similar to the ones given here for capacitor allocations should therefore be applicable. Further research in this area might prove beneficial.

[†] A synchronous condenser can be adjusted to absorb VARS from the system as well as to supply them. In a sense it behaves as a circuit element which is continuously variable between an inductance and a capacitance.

BIBLIOGRAPHY

1. Baldwin, C. J., *et al.*, "A Model for Transmission Planning by Logic," *AIEE Trans.*, vol. *79* Part III, February 1960, pp. 1638-1643.

2. Balinski, M. L., "On Finding Integer Solutions to Linear Programs," *Recent Advances in Operations Research,* The University of Michigan Engineering Summer Conference Notes (1964).

3. Balinski, M. L., "Integer Programming Methods, Uses, Computation," *Recent Mathematical Advances in Operations Research and the Management Sciences,* The University of Michigan Engineering Summer Conference Notes (1965).

4. Baum, W. U. and Frederick W. A., "A Method of Applying Switched and Fixed Capacitors for Voltage Control," *IEEE PAS-84*, January 1965, pp. 42-48.

5. Beale, E. M. L., "Survey of Integer Programming," *Operational Research Quarterly*, vol. *16*, no. 2, June 1965, pp. 219-228.

6. Bloomquist, W. C., *et al.*, *Capacitors for Industry* (New York: John Wiley and Sons, Inc., 1950).

7. Converti, V., *et al.*, "Optimized Distribution and Subtransmission Planning by Digital Computer,"
   Part I - *IEEE PAS-83*, April 1964, pp. 341-347.
   Part II - presented at IEEE Summer General Meeting,
   June 16-21, 1963.
   Part III - *IEEE-PAS-84*, December 1965, pp. 1150-1162.
   Part IV - *IEEE PAS-84*, December 1965, pp. 1165-1177.
   Part V - *IEEE PAS-85*, October 1966, pp. 1083-1091.

8. Cook, R. F., "Optimizing the Application of Shunt Capacitors for Reactive-Volt-Ampere Control and Loss Reduction," *IEEE PAS-80*, August 1961, pp. 430-441.

9. Dantzig, G. B., *Linear Programming and Extensions* (New Jersey: Princeton University Press, 1963).

10. Duran, H., "Optimum Number, Location, and Size of Shunt Capacitors in Radial Distribution Feeders—A Dynamic Programming Approach," *IEEE PAS-87,* September 1968, pp. 1769-1774.

11. Electric Utility Engineering Reference Book, Vol. 3—Distribution Systems. Westinghouse Electric Corporation Staff, 1965, pp. 247-303.

12. Esser, W. F. and Westervelt, F. H., "Investigation of the Application of Heuristic Techniques to Transmission Planning," PICA Proc. 1965, pp. 602-611.

13. Fitzgerald, A. E. and Kingsley, C., Electric Machinery (New York: McGraw Hill Co., 1961 Second Edition).

14. Galloway, C. D. and Garver, L. L., "Computer Design of Single Area Generation Expansions," IEEE PAS-83, April, 1964, pp. 305-311.

15. Garver, L. L., "Power Generation Scheduling by Integer Programming—Development of Theory," IEEE PAS-82, February 1963, pp. 730-735.

16. Gaussens, P. and Calvet, D., "Determining the Operation and Equipment of Electric Power Distribution on Networks," PICA Proc. 1965, pp. 637-657.

17. Jeas, G. and Mitiguy, J. R., "Design of Distribution Circuits by Digital Computer," PICA Proc. 1965, pp. 706-716.

18. Klein, W. A., et al., "Reactive Sources--A Method of Evaluating Their Effectiveness," IEEE PAS-83, July 1964, pp. 714-718.

19. Knight, U. G. W., "The Logical Design of Electrical Networks Using Linear Programming Methods," IEEE Proceedings, vol. 107, Part A(1960), pp. 306-314.

20. Lawler, E. L., "Partitioning Methods for Discrete Optimization Problems," Recent Advances in Operations Research, The University of Michigan Engineering Summer Conferences Notes (1964).

21. Lawler, E. L., "An Introduction to Combinatorial Optimization," Recent Advances in Operations Research, The University of Michigan Engineering Summer Conferences Notes (1964).

22. Lawler, E. L. and Bell, M. D., "A Method for Solving Discrete Optimization Problems," Operations Research, vol. 14, no. 6, November-December 1966, pp. 1098-1112.

23. Lawrence, R. F., *et al.*, "Automated Distribution System Planning," *IEEE PAS-83*, April 1964, pp. 311-316.

24. Maliszewski, R. M., *et al.*, "Linear Programming as an Aid in Planning Kilovar Requirements," *IEEE PAS-87*, December 1968, pp. 1962-1967.

25. Marbury, R. E., *Power Capacitors* (New York: McGraw-Hill, 1949).

26. Montmeat, F. E., *et al.*, "Results of Distribution Expansion Studies by Simulation," presented at the 1969 American Power Conference, Chicago, Illinois (April 22-24, 1969).

27. Nagel, T. J. and Vassell, G. S., "Basic Principle of Planning VAR Control on the American Electric Power System," *IEEE PAS-87*, February 1968, pp. 488-494.

28. Oldfield, J. V. and Lang, T., "The Long Term Design of Electrical Power Distribution Systems," *PICA Proc. 1965,* pp. 671-682.

29. Orth, E. J. and Redmond, J. L., "A Digital Tool for Planning Radial Distribution Systems," *IEEE Trans. on Power Apparatus and Systems—Special Supplement 1963,* pp. 789-798.

30. Peschon, J., *et al.*, "Optimum Control of Reactive Power Flow," *IEEE PAS-87,* January 1968, pp. 40-47.

31. Pierce, John F., "Application of Combinatorial Programming to a Class of All-Zero-One Integer Programming Problems," *Management Science,* vol. *15,* no. 3, November 1968, pp. 191-209.

32. Schmill, J. V., "Optimum Size and Location of Shunt Capacitors on Distribution Feeders," *IEEE PAS-84*, September 1965, pp. 825-832.

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

34. Stevenson, W. D., Jr., *Elements of Power System Analysis* (New York: McGraw-Hill, 1962).

35. Thompson, G. L., "Notes on Heuristic Programming," Recent Advances in Operations Research, The University of Michigan Engineering Summer Conference Notes (1964).

36. Tinney, W. F. and Peschon, J., "On the Use of Power Flow Models for System Planning and Operations," presented at the 1969 American Power Conference, Chicago, Illinois (April 22-24, 1969).

37. Ward, J. B. and Hale, H. W., "Digital Computer Solution of Power-Flow Problems," AIEE Trans., vol. 75 Part III, June 1956, pp. 398-402.

38. Webler, R. M., et al., "Distribution Planning for Load Growth," IEEE PAS-82, December 1963, pp. 908-923.

39. Whysong, J. L., et al., "Computer Program for Automatic Transmission Planning," IEEE PAS-82, February 1963, pp. 774-780.

40. Zaborszky, J. and Rittenhouse, J. W., Electric Power Transmission (New York: Ronald Press, 1954).

Appendix

STATIC CAPACITOR ALLOCATION PROGRAM CODING

SCAP consists of a main program and 17 subprograms. The main program provides the underlying logic of the flow diagram shown in Figure 18 (Chapter 8). The various subprograms perform most of the computation for each of the blocks in this diagram. Three subprograms LF1, LF2 and LF3 perform the operations required in load flow analysis. These were coded by Bill Esser of the Consumers Power Company and are based on one of his programs which has been in use there for a number of years. Two other subprograms, DL and RL, which remove and replace lines and transformers in the load flow data, were also coded by Mr. Esser. All of the other subprograms and the main calling program were coded by the author.

A complete listing of SCAP is provided in the following pages. In order to improve readability, comment statements, which indicate the various operations being performed, have been provided throughout the program. These, in conjunction with the flow charts presented in the text, should simplify the work involved in accomplishing any desired program modifications. As a further aid to program modification, a glossary of the variables that appear is supplied. Since two authors were involved in the coding, it was decided that it would be best to give the glossary in two parts—one for the load flow subprograms and one for the remainder of SCAP. These shall be referred to as the Load Flow Glossary and SCAP Glossary respectively. Some of the variables which appear in the load flow subprograms appear elsewhere in SCAP with a different meaning. Therefore, when look ing up a variable, the appropriate glossary should be consulted. The variables which appear in the load flow subprograms and which are passed to them from the other routines in SCAP through COMMON statements will appear only in the Load Flow Glossary. Thus, if a variable cannot be located in the SCAP glossary, the Load Flow Glossary should be consulted.

The load flow program used in the Static Capacitor Allocation Program was designed to perform the basic calculations of voltage levels and power flows. In the interest of computational speed and simplicity, several common automatic, but time consuming, features found in modern large scale load flow programs were omitted. These include automatic tap changing transformers, tie line control and VAR limits on generators. Three subroutines make up the total program. Subroutine LF1 reads the input data and arranges it into arrays. Subroutine LF2 removes all open lines and lines connected to removed buses and then divides the impedances and power schedules of each constant load bus by the total self admittance of that bus. It then uses a Gauss-Seidel nodal iterative method to calculate the network voltages. The acceleration technique makes use of overcorrection of successive displacements, undercorrection of successive displacements and simultaneous extrapolation of successive displacements[†]. Subroutine LF3 uses the voltages from LF2 and the arrays from LF1 to calculate the power flows. It also handles all output operations.

SCAP is dimensioned to handle systems of up to 100 buses and a maximum of three system states. Application set A can contain up to 20 buses. The load flow subprograms are limited to 100 buses, one swing bus, and 200 transformers and lines.

This appendix is divided into four sections. Section A. 1 gives card formats for SCAP input data. Sections A. 2 and A. 3 are glossarys of SCAP and load flow subprogram variables. Section A. 4 concludes the appendix with a complete listing of the Static Capacitor Allocation Program.

[†] Brown, R. J. and Tinney, W. F., "Digital Solutions for Large Power Networks," AIEE, vol. 76, Part III, pp. 347-351.

A. 1 INPUT CARD FORMATS

SCAP Control Card

Column

2 Number of system states to be considered (3 maximum).

6 Mode of program operation

1 = fixed allocation

2 = switched allocation

3 = fixed and switched allocation.

9-13 Cost of capacitor units per KVAR.

16-20 Cost of switchgear and controls per switched bank.

23-29 Low voltage bound. (x. xxxxx P. U.)

32-38 High voltage bound. (x. xxxxx P. U.)

41-47 Minimum significant bus influence. (x. xxxxx P. U.)

50-57 Capacitor unit size. (x. xxxxxx P. U. susceptance)

60-67 Cost threshold. ($xxxxxxx.) All solutions with cost less than this amount are found. If entry is left blank, only the optimum solution is searched for.

70 Output control variable

0 - Load flow checks made on each solution

1 - Linear approx. voltages are listed.

Outage Case Card

Column

1-3 Terminals of line or transformer to be removed. If no equipment is to be removed, submit blank card for that state.

11-12 Circuit number of line to be removed. (In the case of parallel lines, the user must specify circuit numbers in the load flow data. If there are no parallel lines involved in the outage this entry is left blank.)

Initial Load Flow Control Card

Column

71-80 Case Identification Number

80 Must contain the integer "1"

Load Flow Parameter Card

Column

1-23 Comments

25 Blank if complete case is to be specified, 1 if case is dependent upon the preceding case.

29-32 Base MVA (xxx. x). If left blank, 100. 0 will be assigned by the program

34-42 Convergence Criterion (will be assigned if left blank.)

44-46 Swing Angle (xx. x degrees). (If left blank 45. 0 will be used.)

48-50 Maximum Iteration Count. (If blank, 300 will be assigned.)

71-78 Case Identification Number.

80 Must be blank.

Comment Cards (2)

The next two cards are comment cards. The first 60 lines on each are used. One 120-letter entry is made on the top of each page of output. Column 80 of each card must be blank.

Bus Data Control Card

Column

71-78 Case Identification Number.

80 Must contain the integer "3".

Bus Data Cards

Column

1-8 Bus Name.

12-14 Bus Number.

16-20 Swing Bus Only - Real Component of Voltage (x. xxxx P. U.)

22-26 Swing Bus Only - Imag. component of voltage. (x. xxxx P. U.)

28-32 Watt Generation Schedule (xxx. x MW).

34-38 VAR Generation Schedule (xxx. x MVAR).

40	Bus Type: 1 - P and Q scheduled
	2 - P and V scheduled
	3 - Swing
42-45	Watt Load Schedule (xxx. x MW).
47-50	VAR Load Schedule (xxx. x MVAR).
52-56	Capacitor Bank (x. xxxx P. U. Susceptance)
58-61	Base Voltage (xxx. x KV)
63	Area
65-68	Voltage Schedule for Type 2 Bus (xxx. x KV)
70	Status 0 - Bus to be added
	1 - Bus to be removed
	4 - Card contains changes in bus data.
71-78	Case ID. Number
80	Must be blank.

Line and Transformer Data Control Card

Column	
71-78	Case I. D. Number.
80	Must contain the integer "4".

Line and Transformer Data Cards

Column	
1-23	Comments (descriptions which may be helpful to the user.)
25-27 29-31	Terminals of Equipment (IBUS and JBUS) respectively.)

33-37	Series Resistance (x. xxxx P. U.)	
39-43	Series Reactance (x. xxxx P. U.)	
45-48	(Lines Only) Pi Line Terminal Shunt Conductance (xxx. x%)	
45-48	(Transformers Only) IBUS Tap (xxx. x KV)	
50-53	(Lines Only) Pi Line Terminal Shunt Susceptance (xxx. x%)	
50-53	(Transformers Only) JBUS Tap (xxx. x KV)	
55-58	Branch Rating (xxxx amps. xxx. x MVA)	
60	Rating Code (A - amps, M - MVA)	
62-63	Circuit Number (in case of parallel lines, individual circuit numbers must be assigned by the user.)	
65	Symmetry - (Blank - Sym, 1 - Nonsym.)	
67	Status Code	0 - Branch to be added 1 - Branch to be removed 4 - Card contains changes in branch data.
69	Branch Type	0 - Line 1 - Transformer
71-78	Case Identification	
80	Must be blank.	

Case Termination Control Card

Column

71-78	Case Identification
80	Must contain the integer "7".

Explanation

In the allocation of switched banks, just one set of system loads and generation is considered and this is specified with the card sequence as shown above. In cases where fixed banks may be allocated, two sets of loads, generation and existing capacitor connections are required. The light load conditions are specified first using the same sequence of cards as shown above. Then the heavy load conditions are specified by appending additional load flow control and data cards as follows:

Initial Load Flow Control Card

Load Flow Parameter Card (1 in column 25)

Comment Cards

Bus Data Control Card

Bus Data Cards (4 in column 70 and only data changes given)

Case Termination Card.

If so desired, changes in lines and transformers can also be specified by inserting the appropriate control and data cards, before the Case Termination Card.

A.2 SCAP Glossary

BC	CAPACITØR UNIT SUSCEPTANCE (P.U.)
CAPTR1	STØRAGE FØR EXISTING CAP. BANKS
CAPTR2	STØRAGE FØR EXISTING CAP. BANKS
CF	CØST ØF CAP. UNITS PER KVAR
CØST	CØST ØF ALLØCATIØN X
CØSTM	CURRENT CØST THRESHØLD
CTHRES	CØST THRESHØLD
C1D	NØN-ZERØ ENTRIES ØF DS-MATRIX FØR STATE 1
C2D	NØN-ZERØ ENTRIES ØF DS-MATRIX FØR STATE 2
C3D	NØN-ZERØ ENTRIES ØF DS-MATRIX FØR STATE 3
E1	REAL VØLT. CØMPØNENT IN STATE 1
E2	REAL VØLT. CØMPØNENT IN STATE 2
E3	REAL VØLT. CØMPØNENT IN STATE 3
EH	VØLT. RISE CØNSTRAINT FØR STATE 1
EL1	VØLT. RISE CØNSTRAINT FØR STATE 1
EL2	VØLT. RISE CØNSTRAINT FØR STATE 2
EL3	VØLT. RISE CØNSTRAINT FØR STATE 3
F1	IMAG. VØLT. CØMPØNENT IN STATE 1
F2	IMAG. VØLT. CØMPØNENT IN STATE 2
F3	IMAG. VØLT. CØMPØNENT IN STATE 3
HVL	HIGH VØLTAGE BØUND IN P.U.
ICA	BUSES IN APPLICATIØN SET
ICKT	CIRCUIT NØ. ØF LINE TØ BE REMØVED
ICØN	NØ. ØF STATES TØ BE CØNSIDERED
IENT1	PØSITIØN ØF END ØF CØL. IN C1D
IENT2	PØSITIØN ØF END ØF CØL. IN C2D
IENT3	PØSITIØN ØF END ØF CØL. IN C3D
IMØDE	MØDE ØF PRØGRAM ØPERATIØN
INDM	USED IN SETTING UP INDX
INDX	INDICATES APPLICATIØN SUBSET
INDXS	INDICATES SWITCHED SUBSET
INDMS	USED IN SETTING UP INDXS
INSTL	.T. IF CAPS. INSTALLED AT CØRRESP. BUS
IRW1	RØW PØSITIØN ØF ENTRIES IN C1D
IRW2	RØW PØSITIØN ØF ENTRIES IN C2D
IRW3	RØW PØSITIØN ØF ENTRIES IN C3D
LVL	LØW VØLTAGE BØUND IN P.U.
MCHG	MIN. SIGNIFICANT BUS INFLUENCE
N	HIGHEST BUS NUMBER
NB	NØ. ØF BUSES IN APPLICATIØN SET
NII,NJJ	TERMINALS ØF EQUIPMENT TØ BE FEMØVED
NK	NØ. ØF BUSES IN APPLICATIØN SUBSET
NKK	NØ. ØF BUSES IN SWITCHED SUBSE'
NØLF	ØUTPUT CØNTRØL VARIABLE
RES1	VØLTAGE RISE PREDICTIØN FØR X IN STATE 1
RES2	VØLTAGE RISE PREDICTIØN FØR X IN STATE 2
RES3	VØLTAGE RISE PREDICTIØN FØR X IN STATE 3
SG	CØST ØF SWITCHGEAR & CØNTRØLS
SRES1	VØLTAGE RISE PREDICTIØN FØR XF IN STATE 1

VC1	INITIAL VØLTAGE IN STATE 1
VC2	INITIAL VØLTAGE IN STATE 2
VC3	INITIAL VØLTAGE IN STATE 3
VRGS1	STØRAGE FØR VAR GENERATIØN
VRGS2	STØRAGE FØR VAR GENERATIØN
VRLS1	STØRAGE FØR VAR LØAD
VRLS2	STØRAGE FØR VAR LØAD
WTGS1	STØRAGE FØR WATT GENERATIØN
WTGS2	STØRAGE FØR WATT GENERATIØN
WTLS1	STØRAGE FØR WATT LØAD
WTLS2	STØRAGE FØR WATT LØAD
X	ALLØCATIØN VECTØR
XF	FIXED ALLØCATIØN VECTØR
XM	MAX. NØ. ØF ADDITIØNAL CAP. UNITS
XMAX	MAX.VECTØR IN VECTØR SET

A. 3 Load Flow Glossary

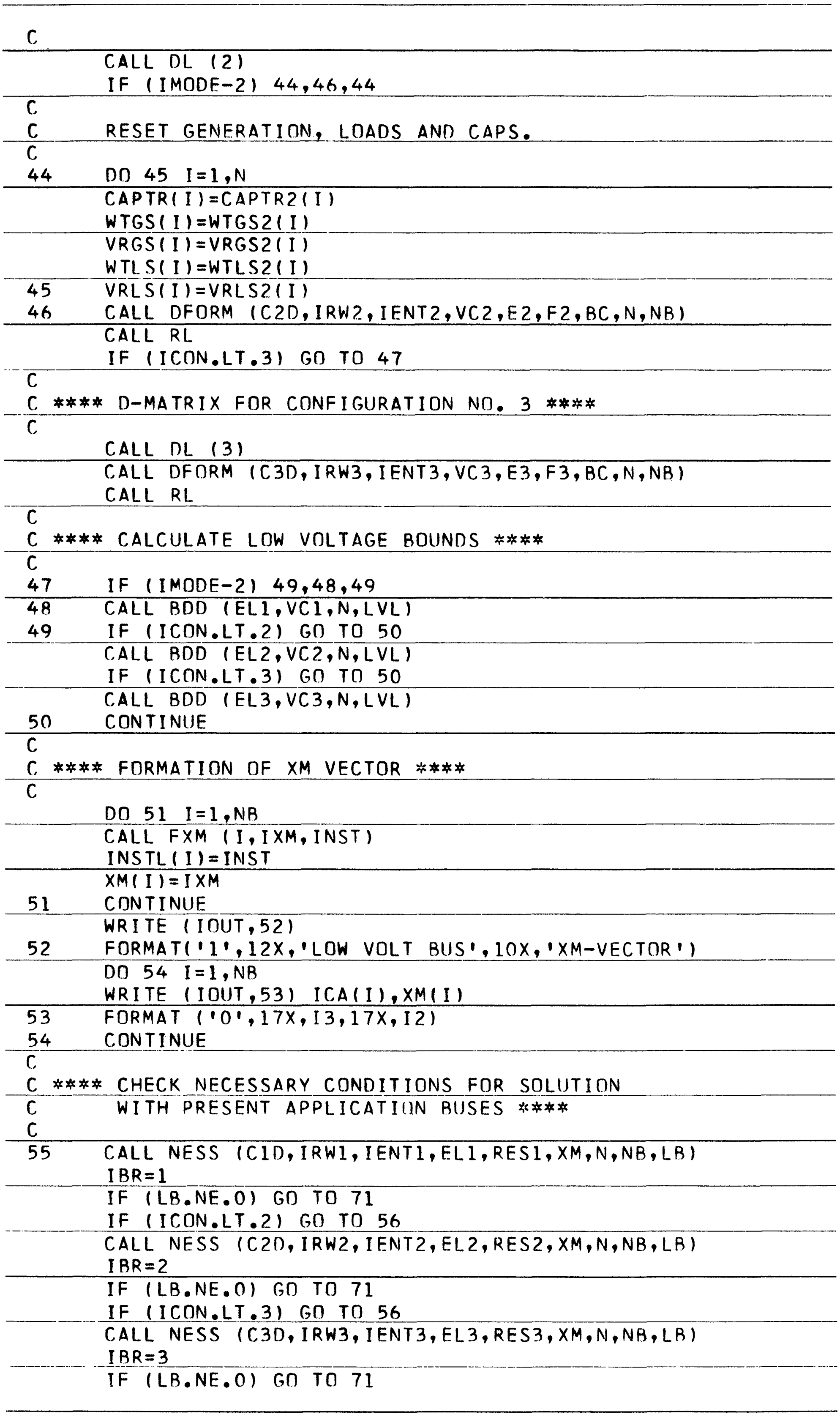
ALPHA	USED IN ACCELERATIØN
ANGLE	ANGLE ØF SWING BUS SØ ØUTPUT CAN RØTATE TØ ZERØ
BASV	BASE VØLTAGE
BETA	USED IN ACCELERATIØN
BII	TØTAL BUS SUSCEPTANCE
BIØ	LINE-SHUNT SUSCEPTANCE ØR TRANSF-TAP ØN JJ BUS
BIØIØ	TEMP FØR BIØ BEFØRE IT GETS ØVERLAID IN LF2
BMVA	BASE MVA
CAPTR	P.U. CAPACITØR SUSCEPTANCE
CØNVG	CØNVERGENCE CRITERIØN
E	REAL CØMPØNENT ØF VØLTAGE
ERRØR	MISMATCH FØR ITERATIØN
E1	P.U. TRANSFØRMER TAP ØN BUS II
E2	P.U. TRANSFØRMER TAP ØN BUS JJ
F	IMAGINARY CØMPØNENT ØF VØLTAGE
GII	TØTAL BUS CØNDUCTANCE
GIØ	LINE-SHUNT CØNDUCTANCE ØR TRANSF-TAP ØN II BUS
GIØIØ	TEMP.FØR GIØ BEFØRE IT GETS ØVERLAID IN LF2
H	REAL CØMPØNENT ØF VØLTAGE CØRRECTIØN
IBRTP	BRANCH TYPE: 0-IF LINE, 1-IF TRANSF
IBT	BRANCH TYPE: 0-IF LINE, 1-IF TRANSF
IBT1	INPUT FØR BUS TYPE
IBUSS	BUS STATUS (I.E. IN ØR ØUT ØF SERVICE)
IBUST	BUS TYPE
ICNTR	INPUT FLAG TØ IDENTIFY DATA FØLLØWING
ICTNØ	CIRCUIT NUMBER
IDENT	CØLUMNS 71 THRØUGH 78 ØN INPUT
II	ØNE TERMINAL BUS FØR A BRANCH
IN	SYMBØLIC TAPE DRIVE-INPUT
INB	HIGHEST BUS NØ.
INL	CØUNT ØF LINES AND TRANSFØRMERS
IØUT	SYMBØLIC TAPE DRIVE-ØUTPUT
IPTR	PØINTER USED FØR SØRTING BRANCHES
IRCTN	INPUT: 0-CØMPLETE DATA, 1-CHANGE DATA
IRTCD	BRANCH LØADING LIMIT CØDE: A=AMPS,M=MVA
IS	SYMMETRY CØDE FØR LINE: 0-SYM., 1-NØT SYM.
ISBN	SWING BUS NUMBER
ISTCD	BRANCH STATUS (I.E. IN ØR ØUT ØF SERVICE)
IS1	DESIGNATES ADDITIØN, CHANGE ØR REMØVAL
ITCM	MAXIMUN NØ.ITERATIØNS-SØLUTIØN QUITS WHEN EXCEEDED
JJ	ØNE TERMINAL BUS FØR A BRANCH
JØ	SWITCH-USE IN ACCELERATIØN
JØT	SWITCH-USE IN ACCELERATIØN
KNAME	CASE TITLE
LI	PRINT LINE CØUNT
ØDH	ACCELERATIØN CØNSTANT
ØDL	ACCELERATIØN CØNSTANT
PL	MW LØSS IN BRANCH

PM	MW MISMATCH ØN BUS
PP	BUS MW & TØTAL BUS ADMITTANCE
QL	MVAR LØSS IN BRANCH
QM	MVAR MISMATCH ØN BUS
QP	BUS MVAR & TØTAL BUS ADMITTANCE
R	BRANCH SERIES RESISTANCE
RATE	BRANCH LØADING LIMIT
RR	TEMP FØR R BEFØRE IT GETS ØVERLAID IN LF2
S	REAL CØMPØNENT ØF CURRENT
S2	VØLTAGE CØRRECTIØNS STØRED FØR ACCELERATIØN
S3	VØLTAGE CØRRECTIØNS STØRED FØR ACCELERATIØN
T	IMAGINARY CØMPØNENT ØF CURRENT
VRGS	MVAR GENERATIØN
VRLS	MVAR LØAD
W	IMAGINARY CØMPØNENT ØF VØLTAGE CØRRECTIØN
WTGS	MW GENERATIØN
WTLS	MW LØAD
X	BRANCH SERIES REACTANCE
XX	TEMP FØR X BEFØRE IT GETS ØVERLAID IN LF2

A.4 Program Listing

```
C                        ####
C
C ****  STATIC CAPACITOR ALLOCATION PROGRAM ****
C
C                        ####
C
C                 100 BUSES MAXIMUM
C                20 APPLICATION BUSES
C                 3 'OUTAGE' CONDITIONS
C
C                        ****
C
C
      DIMENSION RES1(100),RES2(100),RES3(100)
      DIMENSION SRES1(100)
      DIMENSION EL1(100),EL2(100),EL3(100),EH(100)
      DIMENSION C1D(1200),C2D(1200),C3D(1200)
      DIMENSION IRW1(1200),IRW2(1200),IRW3(1200)
      DIMENSION IENT1(20),IENT2(20),IENT3(20)
      DIMENSION VC1(100),VC2(100),VC3(100)
      DIMENSION E(100),F(100),E1(100),F1(100),E2(100),F2(100),
     1 E3(100),F3(100)
      DIMENSION NII(3),NJJ(3),ICKT(3)
      DIMENSION ICA(20)
      DIMENSION WTGS(100),VRGS(100),WTLS(100),VRLS(100)
      DIMENSION WTGS1(100),VRGS1(100),WTLS1(100),VRLS1(100)
      DIMENSION WTGS2(100),VRGS2(100),WTLS2(100),VRLS2(100)
      REAL LVL,HVL,MCHG
      DIMENSION V(100),CAPTR(100),CAPTR1(100),CAPTR2(100)
      DIMENSION PP(100),IBUST(100),QP(100),H(100),S2(100),
     1 GII(100),W(100),S3(100),BII(100),BASV(100),RR(400),
     2 IPTR(400),XX(400),RATE(400),GIOIO(400),BIOIO(400),
     3 P(400),Q(400),IRTCD(400),KNAME(30),IDENT(4),
     4 IN1(100),IN2(100),IBUSS(100)
      INTEGER XM(20),X(20)
      INTEGER XF(20)
      LOGICAL INSTL(20),INST
      LOGICAL CALL
      COMMON /XFC/ XF
      COMMON /INSBK/ INSTL
      COMMON /CHG/ MCHG
      COMMON /LFD/ ICNFG
      COMMON /LINE/ IIB,JJB,RTR,XTX,BIOX,GIOX,ICCT,IBX
      COMMON /LFB/ N
      COMMON /CONF/ ICON
      COMMON /SBIP/ XM,X,EL1,EL2,EL3,EH
      COMMON /SBIP2/ C1D,C2D,C3D,IRW1,IRW2,IRW3,IENT1,IENT2,IENT3
      COMMON /SBIP3/ NB
      COMMON /VLTGE/ V
      COMMON /SC/ CAPTR
      COMMON /LN/ NII,NJJ,ICKT
      COMMON /RESCOM/ RES1,RES2,RES3,SRES1
      COMMON /SCAP/ ICA
      COMMON /MODE/ IMODE,CTHRES
      COMMON /EC/ CF,SG,BC
      COMMON /CAP/ CAPTR1,CAPTR2
      COMMON /LFE/ E,F,ISBN
      COMMON /LF/ PP,WTGS,IBUST,QP,VRGS,H,S2,GII,WTLS,
     1 W,S3,BII,VRLS,BASV,U,RR,IPTR,XX,RATE,GIOIO,
     2 BIOIO,P,Q,IRTCD,KNAME,IDENT,IT,ODL,ODH,ITCM,
      3 ANGLE,IN1,IN2,IBUSS
        IN=5
        IOUT=6
C
C **** DATA INPUT ****
C
        READ (IN,1) ICON,IMODE,CF,SG,LVL,HVL,MCHG,BC,CTHRES,NOLF
1       FORMAT (I2,2X,I2,2(2X,F5.2),3(2X,F7.5),2X,F8.6,2X,F8.0,1X,I2)      1
        READ (IN,2) (NII(I),NJJ(I),ICKT(I),I=1,ICON)
2       FORMAT (I3,2X,I3,2X,I2)
        WRITE (IOUT,3)
3       FORMAT('1'/////////////18X,'## STATIC CAPACITOR'
      1 ,' ALLOCATION ##')
        WRITE (IOUT,4)
4       FORMAT(///22X,'LINE OR TRANSFORMER OUTAGES')
        WRITE (IOUT,5) (I,NII(I),NJJ(I),ICKT(I),I=1,ICON)
5       FORMAT('0',23X,I1,')  LINE ',I3,'-',I3,
      1 '   CKT',I2)
        WRITE (IOUT,6) HVL
6       FORMAT(///'0',16X,'THE MAXIMUM VOLTAGE LEVEL IS '
      1 ,F8.6,' PU')
        WRITE (IOUT,7) LVL
7       FORMAT('0',16X,'THE MINIMUM VOLTAGE LEVEL IS ',
      1 F8.6,' PU')
        WRITE (IOUT,8) BC
8       FORMAT('0',11X,'THE CAPACITOR UNITS ARE OF ',
      1 F8.6,' PU SUSCEPTANCE')
        WRITE (IOUT,9) MCHG
9       FORMAT(///10X,'THE MINIMUM ALLOWABLE VOLTAGE'
      1 ' INCREMENT IS ',F8.6,' PU')
        WRITE (IOUT,10) CF,SG
10      FORMAT('0',8X,'COSTS:   CAPACITOR UNIT $',
      1 F5.2,'/KVAR',3X,'SWITCHGEAR $',F5.2,'K')
        IF (CTHRES.EQ.0.0) GO TO 12
        WRITE (IOUT,11) CTHRES
11      FORMAT('0',9X,'ALL SOLUTIONS WITH COST LESS THAN $',
      1 F8.0,' ARE LISTED')
12      WRITE (IOUT,13) IMODE
13      FORMAT ('0',30X,'MODE=',I2)
C
C **** LOAD FLOW FOR NETWORK CONFIGURATION NO. 1 ****
C       (CONTAINS HIGH VOLT. COND. IF IMODE = 1 OR 3)
C
        CALL LF1
        CALL DL (1)
        CALL SWING (N)
        CALL LF2
        IF (NOLF.EQ.1) GO TO 14
        CALL LF3
14      IF (IMODE-2) 15,18,15
C
C       SAVE GENERATION & LOADING.
C       CALCULATE HIGH VOLTAGE BOUND
C
15      DO 17 I=1,N
        WTGS1(I)=WTGS(I)
        VRGS1(I)=VRGS(I)
        WTLS1(I)=WTLS(I)
        VRLS1(I)=VRLS(I)
      EHT=HVL-V(I)
      IF (EHT.GT.0.0) GO TO 17
      WRITE (IOUT,16) I
 16   FORMAT('OIDEAL CASE VIOLATES HIGH VOLT. BDD AT B#',I3)
      CALL EXIT
 17   EH(I)=EHT
C
C     SAVE VOLTAGES AND CAPS.
C
 18   DO 19 I=1,N
      E1(I)=E(I)
      F1(I)=F(I)
      CAPTR1(I)=CAPTR(I)
      CAPTR2(I)=0.0
 19   VC1(I)=V(I)
      CALL RL
      IF (ICON.LT.2) GO TO 34
C
C **** LOAD FLOW FOR NETWORK CONFIGURATION NO. 2 ****
C
      IF (IMODE-2) 20,22,20
 20   CALL LF1
C
C     SAVE GENERATION, LOADING AND CAPS.
C
      DO 21 I=1,N
      CAPTR2(I)=CAPTR(I)
      WTGS2(I)=WTGS(I)
      VRGS2(I)=VRGS(I)
      WTLS2(I)=WTLS(I)
 21   VRLS2(I)=VRLS(I)
 22   CALL DL (2)
      CALL SWING (N)
      CALL LF2
      IF (NOLF.EQ.1) GO TO 23
      CALL LF3
C
C     SAVE VOLTAGES
C
 23   DO 24 I=1,N
      E2(I)=E(I)
      F2(I)=F(I)
 24   VC2(I)=V(I)
      CALL RL
      IF (ICON.LT.3) GO TO 34
C
C **** LOAD FLOW FOR NETWORK CONFIGURATION NO. 3 ****
C
      CALL DL (3)
      CALL SWING (N)
      CALL LF2
      IF (NOLF.EQ.1) GO TO 25
      CALL LF3
C
C     SAVE VOLTAGES
C
 25   DO 26 I=1,N
      E3(I)=E(I)
      F3(I)=F(I)
 26   VC3(I)=V(I)
      CALL RL
      IF (NOLF.NE.1) GO TO 34
```

```
C
C      # SIMPLIFIED VOLTAGE OUTPUT #
C
       WRITE (IOUT,27)
 27    FORMAT('1',15X,'*** EXISTING VOLTAGES ***')
       WRITE (IOUT,28)
 28    FORMAT('0',2X,'STATE #1',12X,'STATE #2',12X,'STATE #3')
       WRITE (IOUT,29)
 29    FORMAT('0BUS',3X,'PU VOLT.',6X,'BUS',3X,'PU VOLT.',
      1 6X,'BUS',3X,'PU VOLT.')
       DO 33 I=1,N
       WRITE (IOUT,30) I,VC1(I)
 30    FORMAT ('0',I3,4X,F5.3)
       IF (ICON.LT.2) GO TO 33
       WRITE (IOUT,31) I,VC2(I)
 31    FORMAT ('+',20X,I3,4X,F5.3)
       IF (ICON.LT.3) GO TO 33
       WRITE (IOUT,32) I,VC3(I)
 32    FORMAT ('+',40X,I3,4X,F5.3)
 33    CONTINUE
C
C **** PUT LOW VOLTAGE BUSES IN APPLICATION SET ****
C
 34    NB=0
       DO 36 I=1,N
       IF (NB.GT.20) GO TO 38
       IF (VC1(I).LT.LVL) GO TO 35
       IF (ICON.LT.2) GO TO 36
       IF (VC2(I).LT.LVL) GO TO 35
       IF (ICON.LT.3) GO TO 36
       IF (VC3(I).GE.LVL) GO TO 36
 35    NB=NB+1
       ICA(NB)=I
 36    CONTINUE
       IF (NB.NE.0) GO TO 40
       WRITE (IOUT,37)
 37    FORMAT('0NO LOW VOLTAGE BUSES PRESENT - SYSTEM OK')
       CALL EXIT
 38    WRITE (IOUT,39)
 39    FORMAT('0TOO MANY LOW VOLTAGE BUSES')
       CALL EXIT
C
C **** D-MATRIX FOR CONFIGURATION NO. 1 ****
C
 40    CALL DL (1)
       IF (IMODE-2) 41,43,41
C
C      RESET GENERATION, LOADS AND CAPS.
C
 41    DO 42 I=1,N
       CAPTR(I)=CAPTR1(I)
       WTGS(I)=WTGS1(I)
       VRGS(I)=VRGS1(I)
       WTLS(I)=WTLS1(I)
 42    VRLS(I)=VRLS1(I)
 43    CALL DFORM (C1D,IRW1,IENT1,VC1,E1,F1,BC,N,NB)
       CALL RL
       IF (ICON.LT.2) GO TO 47
C
C **** D-MATRIX FOR CONFIGURATION NO. 2 ****
```

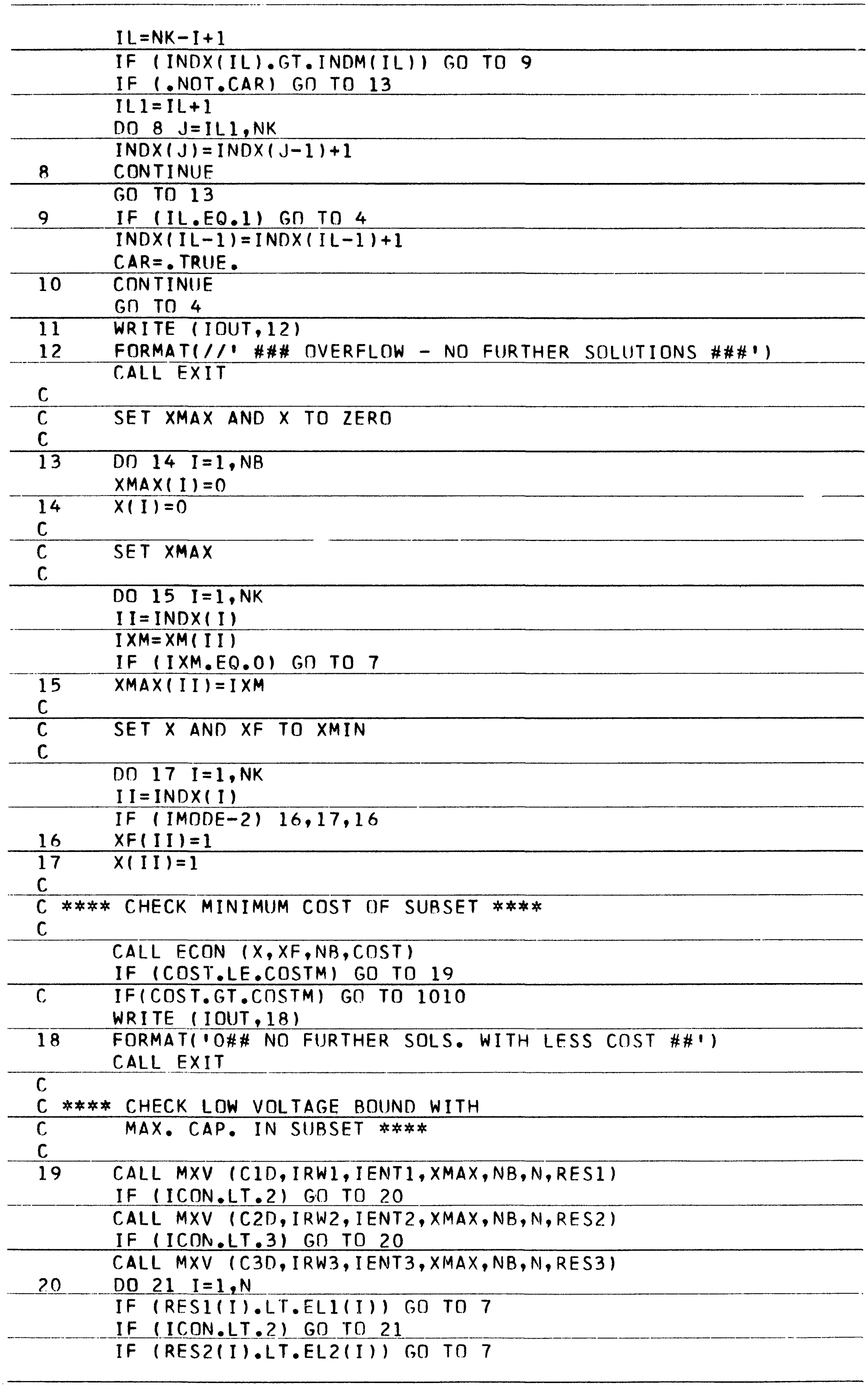


```
C
      CALL DL (2)
      IF (IMODE-2) 44,46,44
C
C     RESET GENERATION, LOADS AND CAPS.
C
44    DO 45 I=1,N
      CAPTR(I)=CAPTR2(I)
      WTGS(I)=WTGS2(I)
      VRGS(I)=VRGS2(I)
      WTLS(I)=WTLS2(I)
45    VRLS(I)=VRLS2(I)
46    CALL DFORM (C2D,IRW2,IENT2,VC2,E2,F2,BC,N,NB)
      CALL RL
      IF (ICON.LT.3) GO TO 47
C
C **** D-MATRIX FOR CONFIGURATION NO. 3 ****
C
      CALL DL (3)
      CALL DFORM (C3D,IRW3,IENT3,VC3,E3,F3,BC,N,NB)
      CALL RL
C
C **** CALCULATE LOW VOLTAGE BOUNDS ****
C
47    IF (IMODE-2) 49,48,49
48    CALL BDD (EL1,VC1,N,LVL)
49    IF (ICON.LT.2) GO TO 50
      CALL BDD (EL2,VC2,N,LVL)
      IF (ICON.LT.3) GO TO 50
      CALL BDD (EL3,VC3,N,LVL)
50    CONTINUE
C
C **** FORMATION OF XM VECTOR ****
C
      DO 51 I=1,NB
      CALL FXM (I,IXM,INST)
      INSTL(I)=INST
      XM(I)=IXM
51    CONTINUE
      WRITE (IOUT,52)
52    FORMAT('1',12X,'LOW VOLT BUS',10X,'XM-VECTOR')
      DO 54 I=1,NB
      WRITE (IOUT,53) ICA(I),XM(I)
53    FORMAT ('0',17X,I3,17X,I2)
54    CONTINUE
C
C **** CHECK NECESSARY CONDITIONS FOR SOLUTION
C       WITH PRESENT APPLICATION BUSES ****
C
55    CALL NESS (C1D,IRW1,IENT1,EL1,RES1,XM,N,NB,LB)
      IBR=1
      IF (LB.NE.0) GO TO 71
      IF (ICON.LT.2) GO TO 56
      CALL NESS (C2D,IRW2,IENT2,EL2,RES2,XM,N,NB,LB)
      IBR=2
      IF (LB.NE.0) GO TO 71
      IF (ICON.LT.3) GO TO 56
      CALL NESS (C3D,IRW3,IENT3,EL3,RES3,XM,N,NB,LB)
      IBR=3
      IF (LB.NE.0) GO TO 71
```

```
C
C **** CALL IMPLICIT ENUMERATION ALGORITHM ****
C
56    CALL IEA
      IF (NOLF.EQ.1) GO TO 65
C
C     # LOAD FLOW OUTPUT #
C
57    CALL INITL (E1,F1,N)
      IF (IMODE-2) 58,60,58
C
C     RESET GENERATION, LOADING AND CAPS.
C
58    DO 59 I=1,N
      CAPTR(I)=CAPTR1(I)
      WTGS(I)=WTGS1(I)
      VRGS(I)=VRGS1(I)
      WTLS(I)=WTLS1(I)
59    VRLS(I)=VRLS1(I)
      IF (IMODE.EQ.1) GO TO 60
      CALL CAPCHK (XF,1,VC1,SRES1,NB,N,BC)
      GO TO 61
60    CALL CAPCHK (X,1,VC1,RES1,NB,N,BC)
61    IF (ICON.LT.2) GO TO 70
      IF (IMODE-2) 62,64,62
C
C     RESET GENERATION, LOADING AND CAPS.
C
62    DO 63 I=1,N
      CAPTR(I)=CAPTR2(I)
      WTGS(I)=WTGS2(I)
      VRGS(I)=VRGS2(I)
      WTLS(I)=WTLS2(I)
63    VRLS(I)=VRLS2(I)
64    CALL INITL (E2,F2,N)
      CALL CAPCHK (X,2,VC2,RES2,NB,N,BC)
      IF (ICON.LT.3) GO TO 70
      CALL INITL (E3,F3,N)
      CALL CAPCHK (X,3,VC3,RES3,NB,N,BC)
      GO TO 70
C
C     # APPROX. VOLTAGE OUTPUT #
C
65    WRITE (IOUT,66)
66    FORMAT('1',9X,'** APPROX. CORRECTED VOLTAGES **')
      WRITE (IOUT,28)
      WRITE (IOUT,29)
      DO 69 I=1,N
      IF (IMODE.EQ.1) GO TO 67
      V1=VC1(I)+RES1(I)
      GO TO 68
67    V1=VC1(I)+SRES1(I)
68    WRITE (IOUT,30) I,V1
      IF (ICON.LT.2) GO TO 69
      V2=VC2(I)+RES2(I)
      WRITE (IOUT,31) I,V2
      IF (ICON.LT.3) GO TO 69
      V3=VC3(I)+RES3(I)
      WRITE (IOUT,32) I,V3
69    CONTINUE
```

```
C
C        ENUMERATION CONTINUED
C
 70     CALL CONTIE
        IF (NOLF.EQ.1) GO TO 65
        GO TO 57
C
C **** ADDITION OF BUS TO APPLICATION SET ****
C
C       DETERMINE BUS WITH MAX. INFLUENCE
C
 71     IF (IBR-2) 72,73,74
 72     CALL INFL (C1D,IRW1,IENT1,VC1,LB,ICA,NB)
        GO TO 75
 73     CALL INFL (C2D,IRW2,IENT2,VC2,LB,ICA,NB)
        GO TO 75
 74     CALL INFL (C3D,IRW3,IENT3,VC3,LB,ICA,NB)
C
C        UPDATE LINEAR MODELS
C
 75     NB=NB+1
        ICA(NB)=LB
        DO 76 I=1,N
        WTGS(I)=WTGS1(I)
        VRGS(I)=VRGS1(I)
        WTLS(I)=WTLS1(I)
        VRLS(I)=VRLS1(I)
 76     CAPTR(I)=CAPTR1(I)
        CAPTR(LB)=CAPTR(LB)+BC
        CALL DL (1)
        CALL CADD (C1D,IRW1,IENT1,NB,N,BC,VC1,E1,F1)
        CALL RL
        IF (ICON.LT.2) GO TO 78
        DO 77 I=1,N
        WTGS(I)=WTGS2(I)
        VRGS(I)=VRGS2(I)
        WTLS(I)=WTLS2(I)
        VRLS(I)=VRLS2(I)
 77     CAPTR(I)=CAPTR2(I)
        CALL DL (2)
        CALL CADD (C2D,IRW2,IENT2,NB,N,BC,VC2,E2,F2)
        CALL RL
        IF (ICON.LT.3) GO TO 78
        CALL DL (3)
        CALL CADD (C3D,IRW3,IENT3,NB,N,BC,VC3,E3,F3)
        CALL RL
 78     CAPTR(LB)=CAPTR(LB)-BC
        CALL FXM (NB,IXM,INST)
        XM(NB)=IXM
        INSTL(NB)=INST
        WRITE (IOUT,79) ICA(NB),XM(NB)
 79     FORMAT('0## BUS NO.',I4,' WITH XM =',I3,' ADDED ##')
        GO TO 55
        END
C
C
C
        SUBROUTINE IEA
C
C ****************************************
```

```
C *                                         *
C **** IMPLICIT ENUMERATION ALGORITHM ####
C *                                         *
C *******************************************
C
      DIMENSION RES1(100),RES2(100),RES3(100)
      DIMENSION SRES1(100)
      INTEGER INDX(20),INDM(20),XMAX(20),P
      INTEGER XF(20),INDXS(20),INDMS(20)
      LOGICAL CAR
      LOGICAL  FS,CARY,OVER,LV1
      LOGICAL SCAR,FESW
      LOGICAL THOLD
      DIMENSION EL1(100),EL2(100),EL3(100)
      DIMENSION EH(100)
      DIMENSION C1D(1200),C2D(1200),C3D(1200)
      DIMENSION IRW1(1200),IRW2(1200),IRW3(1200)
      DIMENSION IENT1(20),IENT2(20),IENT3(20)
      INTEGER XM(20),X(20),ICA(20)
      COMMON /LFB/ N
      COMMON/CONF/ ICON
      COMMON /RESCOM/ RES1,RES2,RES3,SRES1
      COMMON /SCAP/ ICA
      COMMON /SBIP/ XM,X,EL1,EL2,EL3,EH
      COMMON /SBIP2/ C1D,C2D,C3D,IRW1,IRW2,IRW3,IENT1,IENT2,IENT3
      COMMON /SBIP3/ NB
      COMMON /MODE/ IMODE,CTHRES
      COMMON /EC/ CF,SG,BC
      COMMON /XFC/ XF
      IN=5
      IOUT=6
C
C ****SET STARTING COST ****
C
C     SET XF TO ZERO
C
      DO 1 I=1,NB
1     XF(I)=0
      THOLD=.FALSE.
      IF (CTHRES.NE.0.0) THOLD=.TRUE.
      IF (.NOT.THOLD) GO TO 2
      COSTM=CTHRES
      GO TO 3
2     CALL ECON (XM,XF,NB,COSTM)
C
C **** FORM APPLICATION SUBSET ****
C
3     NK=0
4     NK=NK+1
      IF (NK.GT.NB) GO TO 11
      DO 5 I=1,NK
      IZ=NB-I+1
      IX=NK-I+1
5     INDM(IX)=IZ
      DO 6 I=1,NK
6     INDX(I)=I
      GO TO 13
7     CAR=.FALSE.
      INDX(NK)=INDX(NK)+1
      DO 10 I=1,NK
```



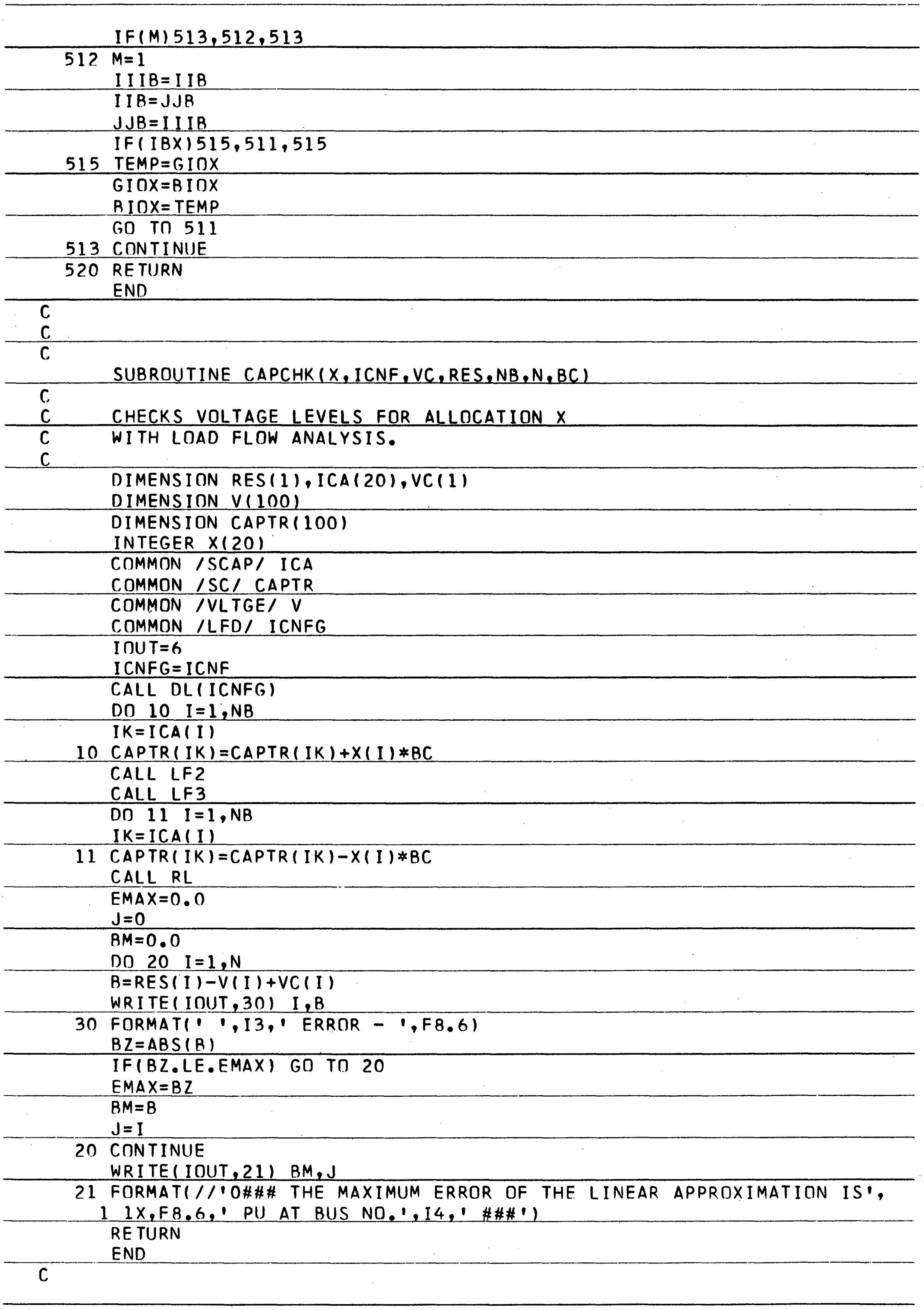
```
      IL=NK-I+1
      IF (INDX(IL).GT.INDM(IL)) GO TO 9
      IF (.NOT.CAR) GO TO 13
      IL1=IL+1
      DO 8 J=IL1,NK
      INDX(J)=INDX(J-1)+1
8     CONTINUE
      GO TO 13
9     IF (IL.EQ.1) GO TO 4
      INDX(IL-1)=INDX(IL-1)+1
      CAR=.TRUE.
10    CONTINUE
      GO TO 4
11    WRITE (IOUT,12)
12    FORMAT(//' ### OVERFLOW - NO FURTHER SOLUTIONS ###')
      CALL EXIT
C
C     SET XMAX AND X TO ZERO
C
13    DO 14 I=1,NB
      XMAX(I)=0
14    X(I)=0
C
C     SET XMAX
C
      DO 15 I=1,NK
      II=INDX(I)
      IXM=XM(II)
      IF (IXM.EQ.0) GO TO 7
15    XMAX(II)=IXM
C
C     SET X AND XF TO XMIN
C
      DO 17 I=1,NK
      II=INDX(I)
      IF (IMODE-2) 16,17,16
16    XF(II)=1
17    X(II)=1
C
C **** CHECK MINIMUM COST OF SUBSET ****
C
      CALL ECON (X,XF,NB,COST)
      IF (COST.LE.COSTM) GO TO 19
C     IF(COST.GT.COSTM) GO TO 1010
      WRITE (IOUT,18)
18    FORMAT('0## NO FURTHER SOLS. WITH LESS COST ##')
      CALL EXIT
C
C **** CHECK LOW VOLTAGE BOUND WITH
C      MAX. CAP. IN SUBSET ****
C
19    CALL MXV (C1D,IRW1,IENT1,XMAX,NB,N,RES1)
      IF (ICON.LT.2) GO TO 20
      CALL MXV (C2D,IRW2,IENT2,XMAX,NB,N,RES2)
      IF (ICON.LT.3) GO TO 20
      CALL MXV (C3D,IRW3,IENT3,XMAX,NB,N,RES3)
20    DO 21 I=1,N
      IF (RES1(I).LT.EL1(I)) GO TO 7
      IF (ICON.LT.2) GO TO 21
      IF (RES2(I).LT.EL2(I)) GO TO 7
      IF (ICON.LT.3) GO TO 21
      IF (RES3(I).LT.EL3(I)) GO TO 7
21    CONTINUE
      GO TO 35
C
C **** ADDITION OF 1 TO X-VECTOR ****
C
22    X(NB)=X(NB)+1
      DO 25 I=1,NB
      IL=NB-I+1
      IF (X(IL).LE.XMAX(IL)) GO TO 34
      IF (IL.EQ.1) GO TO 7
      IF (XMAX(IL).NE.0) GO TO 23
      X(IL)=0
      GO TO 24
23    X(IL)=1
24    X(IL-1)=X(IL-1)+1
25    CONTINUE
      GO TO 7
C
C **** SKIP OPERATION ****
C
      ENTRY CONTIE
      IF (NB.EQ.1) GO TO 11
      IF (NKK.NE.0) GO TO 47
26    DO 33 I=1,NB
      IL=NB-I+1
      IF (X(IL).EQ.0) GO TO 32
      IF (X(IL).EQ.1) GO TO 32
      IF (XMAX(IL).NE.0) GO TO 27
      X(IL)=0
      GO TO 28
27    X(IL)=1
28    X(IL-1)=X(IL-1)+1
      II=I+1
      DO 31 J=II,NB
      IE=NB-J+1
      IF (X(IE).LE.XMAX(IE)) GO TO 34
      IF (IE.EQ.1) GO TO 7
      IF (XMAX(IE).NE.0) GO TO 29
      X(IE)=0
      GO TO 30
29    X(IE)=1
30    X(IE-1)=X(IE-1)+1
31    CONTINUE
32    IF (IL.EQ.2) GO TO 7
33    CONTINUE
C
C **** COST OF ALLOCATION X ****
C
34    CALL ECON (X,XF,NB,COST)
      IF (COST.GT.COSTM) GO TO 26
C
C **** CONSTRAINT BASED DECISIONS ****
C
35    CALL MXV (C1D,IRW1,IENT1,X,NB,N,RES1)
      IF (ICON.LT.2) GO TO 36
      CALL MXV (C2D,IRW2,IENT2,X,NB,N,RES2)
      IF (ICON.LT.3) GO TO 36
      CALL MXV (C3D,IRW3,IENT3,X,NB,N,RES3)
      IF (ICON.LT.3) GO TO 21
      IF (RES3(I).LT.EL3(I)) GO TO 7
 21   CONTINUE
      GO TO 35
C
C **** ADDITION OF 1 TO X-VECTOR ****
C
 22   X(NB)=X(NB)+1
      DO 25 I=1,NB
      IL=NB-I+1
      IF (X(IL).LE.XMAX(IL)) GO TO 34
      IF (IL.EQ.1) GO TO 7
      IF (XMAX(IL).NE.0) GO TO 23
      X(IL)=0
      GO TO 24
 23   X(IL)=1
 24   X(IL-1)=X(IL-1)+1
 25   CONTINUE
      GO TO 7
C
C **** SKIP OPERATION ****
C
      ENTRY CONTIE
      IF (NB.EQ.1) GO TO 11
      IF (NKK.NE.0) GO TO 47
 26   DO 33 I=1,NB
      IL=NB-I+1
      IF (X(IL).EQ.0) GO TO 32
      IF (X(IL).EQ.1) GO TO 32
      IF (XMAX(IL).NE.0) GO TO 27
      X(IL)=0
      GO TO 28
 27   X(IL)=1
 28   X(IL-1)=X(IL-1)+1
      II=I+1
      DO 31 J=II,NB
      IE=NB-J+1
      IF (X(IE).LE.XMAX(IE)) GO TO 34
      IF (IE.EQ.1) GO TO 7
      IF (XMAX(IE).NE.0) GO TO 29
      X(IE)=0
      GO TO 30
 29   X(IE)=1
 30   X(IE-1)=X(IE-1)+1
 31   CONTINUE
 32   IF (IL.EQ.2) GO TO 7
 33   CONTINUE
 34   IF (IMODE-2) 74,72,74
 74   DO 70 I=1,NB
 70   XF(I)=X(I)
C
C **** COST OF ALLOCATION X ****
C
 72   CALL ECON (X,XF,NB,COST)
      IF (COST.GT.COSTM) GO TO 26
C
C **** CONSTRAINT BASED DECISIONS ****
C
 35   CALL MXV (C1D,IRW1,IENT1,X,NB,N,RES1)
      IF (ICON.LT.2) GO TO 36
      CALL MXV (C2D,IRW2,IENT2,X,NB,N,RES2)
      IF (ICON.LT.3) GO TO 36
      CALL MXV (C3D,IRW3,IENT3,X,NB,N,RES3)
```

```
C
      CALL ECON (X,XF,NB,COST)
      IF (COST.GT.COSTM) GO TO 26
      CALL MXV (C1D,IRW1,IENT1,XF,NB,N,SRES1)
      DO 54 I=1,N
      IF (SRES1(I).GT.EH(I)) GO TO 47
54    CONTINUE
      FESW=.TRUE.
C
C  **** FEASIBLE SOLUTION ****
C
55    WRITE (IOUT,56) COST
56    FORMAT('1'////////10X,'## FEASIBLE SOLUTION:'
     1 ' COST = $',F8.0,'K  ##')
      DO 61 I=1,NB
      NN=X(I)
      IF (NN.EQ.0) GO TO 61
      WRITE (IOUT,57) ICA(I),NN
57    FORMAT('0',15X,'BUS NO.',I4,':',I3,
     1 '  CAPACITOR UNITS')
      IF (XF(I).EQ.0) GO TO 59
      WRITE (IOUT,58)
58    FORMAT ('+',30X,'F')
      GO TO 61
59    WRITE (IOUT,60)
60    FORMAT ('+',30X,'S')
61    CONTINUE
      IF (THOLD) GO TO 62
      COSTM=COST
62    RETURN
      END
C
C
C
      SUBROUTINE FXM (I,IXM,INST)
C
C     GENERATES I-TH ELEMENT OF XM-VECTOR AND
C     RETURNS IT IN IXM.  INST IS .TRUE. IF CAPS.
C     ARE ALREADY INSTALLED AT BUS ICA(I).
C
      REAL MCHG
      LOGICAL INST
      DIMENSION CAPTR1(100),CAPTR2(100)
      DIMENSION ICA(20),C1D(1200),C2D(1200),C3D(1200)
      DIMENSION IRW1(1200),IRW2(1200),IRW3(1200)
      DIMENSION IENT1(20),IENT2(20),IENT3(20)
      COMMON /EC/ CF,CSW,BC
      COMMON /SCAP/ ICA /CHG/ MCHG /CONF/ ICON
      COMMON /LFB/ N
      COMMON /SBIP2/ C1D,C2D,C3D,.     IRW2,IRW3,IENT1,IENT2,IENT3
      COMMON /CAP/ CAPTR1,CAPTR2
      III=ICA(I)
      INST=.FALSE.
      C1=CAPTR1(III)
      C2=CAPTR2(III)
      D=AMAX1(C1,C2)
      IF (D.NE.0.0) INST=.TRUE.
      D2=0.0
      D3=0.0
      CALL DFIND (I,III,D1,C1D,IRW1,IENT1)
      IF (ICON.LT.2) GO TO 1
      CALL DFIND (I,III,D2,C2D,IRW2,IENT2)
      IF (ICON.LT.3) GO TO 1
      CALL DFIND (I,III,D3,C3D,IRW3,IENT3)
 1    DI=AMAX1(D1,D2,D3)
      IF (DI.LT.MCHG) GO TO 2
      DX=.045/DI
      IXM=INT(DX)
      DZ=(D+.01)/BC
      INSTL=INT(DZ)
      IXM=IXM-INSTL
      IF (IXM.LT.0) IXM=0
      GO TO 3
 2    IXM=0
 3    RETURN
      END
C
C
C
      SUBROUTINE INFL (CD,IRW,IENT,VC,LB,ICA,NB)
C
C     FINDS BUS AT WHICH CAPACITOR ADDITIONS WILL
C     GIVE MAXIMUM AID TO BUS LB.  THE NUMBER OF
C     THIS BUS IS RETURNED IN VARIABLE LB.
C
      DIMENSION CD(1),IRW(1),IENT(1),VC(1),ICA(1)
      COMMON /CHG/ MCHG
      IOUT=6
      DO 1 I=1,NB
      IF (ICA(I).NE.LB) GO TO 1
      IU=I
      GO TO 2
 1    CONTINUE
 2    IL=IU-1
      IL=IENT(IL)+1
      IU=IENT(IU)
      BMAX=0.0
      LC=0
      DO 4 I=IL,IU
      IK=IRW(I)
      DO 3 J=1,NB
      IF (IK.EQ.ICA(J)) GO TO 4
 3    CONTINUE
      BI=CD(I)*VC(IK)/VC(LB)
      IF (BI.LE.BMAX) GO TO 4
      BMAX=BI
      LC=IK
 4    CONTINUE
      LB=LC
      IF (BMAX.GE.MCHG) GO TO 6
      WRITE (IOUT,5) LB
 5    FORMAT('0** BUS#',I3,' CANNOT BE BROUGHT TO MIN. VOLTAGE BY CAPS')
 6    RETURN
      END
C
C
C
      SUBROUTINE NESS (CD,IRW,IENT,EL,RES,XM,N,NB,LB)
C
C     CHECKS TO SEE IF LOWER BDD. CAN BE SATISFIED
```

```
C       WITH EXISTING APPLICATION SET.  IF IT CAN LB=0.
C       IF IT CAN'T THE NUMBER OF THE LOWEST VOLTAGE
C       BUS REMAINING AFTER CAPACITOR ADDITIONS IS
C       RETURNED IN LB.
C
        DIMENSION CD(1),EL(1),RES(1)
        DIMENSION IRW(1),IENT(1)
        INTEGER XM(1)
        LB=0
        CALL MXV (CD,IRW,IENT,XM,NB,N,RES)
        DIFM=0.0
        DO 1 I=1,N
        IF (RES(I).GE.EL(I)) GO TO 1
        DIF=EL(I)-RES(I)
        IF (DIF.LT.DIFM) GO TO 1
        DIFM=DIF
        LB=I
1       CONTINUE
        RETURN
        END
C
C
C
        SUBROUTINE BDD (EL,VC,N,LVL)
C
C       COMPUTES LOWER BOUND FOR USE IN IEA
C
        DIMENSION EL(1),VC(1)
        REAL LVL
        DO 1 I=1,N
1       EL(I)=LVL-VC(I)
        RETURN
        END
C
C
C
        SUBROUTINE MXV (CD,IRW,IENT,IX,NB,N,R)
C
C       FORMS PRODUCT OF DS-MATRIX AND VECTOR X.
C       RESULT IS PLACED IN VECTOR R.  DS-MATRIX
C       IS STORED IN 3 ARRAYS - CD, IRW AND IENT.
C       NB IS NO. OF COLS. IN DS, N IS NO. OF ROWS.
C
        DIMENSION CD(1),IRW(1),IENT(1),IX(1),R(1)
        DO 1 I=1,N
1       R(I)=0.0
        IC=1
        IL=1
2       IU=IENT(IC)
        IIX=IX(IC)
        IF (IIX.EQ.0) GO TO 4
        DO 3 J=IL,IU
        JJ=IRW(J)
3       R(JJ)=R(JJ)+CD(J)*IIX
4       IL=IU+1
        IC=IC+1
        IF (IC.LE.NB) GO TO 2
        RETURN
        END
C
```

```
C
C
      SUBROUTINE DFORM (CD,IRW,IENT,VC,EI,FI,BC,N,NB)
C
C     CONSTRUCTS D-MATRIX AND STORES IT USING SPARSE MATRIX
C     TECHNIQUES IN ARRAYS CD, IRW AND IENT
C
      DIMENSION CAPTR(100),E(100),F(100),ICA(20),VC(1),CD(1)
      DIMENSION IRW(1),IENT(1)
      DIMENSION EI(1),FI(1)
      REAL MCHG
      COMMON /CHG/ MCHG /LFE/ E,F,ISBN
      COMMON /SC/ CAPTR /SCAP/ ICA
      LOGICAL CALL,ADD
      IOUT=6
      CALL=.FALSE.
      ADD=.FALSE.
      NE=0
      GO TO 1
      ENTRY CADD(CD,IRW,IENT,NB,N,BC,VC,EI,FI)
      ADD=.TRUE.
      II=NB-1
      NE=IENT(II)
      I=NB
      GO TO 2
1     DO 8 I=1,NB
      IK=ICA(I)
      CAPTR(IK)=CAPTR(IK)+2*BC
2     CALL INITL (EI,FI,N)
      CALL LF2
      DO 7 J=1,N
      V2=E(J)**2+F(J)**2
      B=(SQRT(V2)-VC(J))*.5
      BZ=ABS(B)
      IF (BZ.LE.MCHG) GO TO 7
      IF (B.GE.0.0) GO TO 6
      WRITE (IOUT,3) J
3     FORMAT('ONEG. INCREMENT AT BUS #',I3)
      CALL=.TRUE.
      GO TO 7
4     WRITE (IOUT,5)
5     FORMAT('OD-MATRIX OVERFLOW-INCREASE DIMENSIONS')
      CALL EXIT
6     NE=NE+1
      IF (NE.GT.1200) GO TO 4
      CD(NE)=B
      IRW(NE)=J
7     CONTINUE
      IENT(I)=NE
      IF (ADD) GO TO 9
      CAPTR(IK)=CAPTR(IK)-2*BC
8     CONTINUE
9     IF (CALL) CALL EXIT
      RETURN
      END
C
C
C
      SUBROUTINE DFIND (I,IR,D,CD,IRW,IENT)
C
```

```
C     RETURNS ELEMENT OF D-MATRIX IN IR-TH ROW
C     AND I-TH COLUMN.
C
      DIMENSION CD(1),IRW(1),IENT(1)
      IF (I.NE.1) GO TO 1
      IL=1
      GO TO 2
1     IL=IENT(I-1)+1
2     IU=IENT(I)
      DO 3 J=IL,IU
      IF (IRW(J).NE.IR) GO TO 3
      D=CD(J)
      GO TO 4
3     CONTINUE
4     RETURN
      END
C
C
C
      SUBROUTINE INITL (EJ,FJ,N)
C
C     INITIALIZES VOLTAGE VECTORS OF LOAD FLOW
C     PROGRAM TO VOLTAGE FOR THE APPROPRIATE
C     OUTAGE CASE.
C
      DIMENSION E(100),F(100),EJ(1),FJ(1)
      COMMON /LFE/ E,F,ISBN
      DO 1 I=1,N
      E(I)=EJ(I)
1     F(I)=FJ(I)
      RETURN
      END
C
C
C
      SUBROUTINE SWING (N)
C
C     INITIALIZES VOLTAGE VECTORS OF LOAD FLOW
C     PROGRAM TO SWING MACHINE VALUE.
C
      DIMENSION E(100),F(100)
      COMMON /LFE/ E,F,ISBN
      DO 1 I=1,N
      E(I)=E(ISBN)
1     F(I)=F(ISBN)
      RETURN
      END
C
C
C
      SUBROUTINE DL(K)
C
C     REMOVES EQUIPMENT BETWEEN TERMINALS
C     NNI(K) AND NJJ(K) WITH CIRCUIT
C     NUMBER ICKT(K).
C
      DIMENSION NII(3),NJJ(3),ICKT(3)
      DIMENSION II(400),JJ(400),R(400),X(400),GIO(400),BIO(400)
      DIMENSION ICTNO(400),IBRTP(400),ISTCD(400)
      COMMON /LINE/ IIB,JJB,RTR,XTX,BIOX,GIOX,ICCT,IBX
      COMMON /LN/ NII,NJJ,ICKT
      COMMON /LFC/ II,JJ,R,X,GIO,BIO,ICTNO,IBRTP,ISTCD
      COMMON /LFA/ INL
      IOUT=6
      IIB=NII(K)
      IF(IIB.EQ.0) GO TO 5
      JJB=NJJ(K)
      ICCT=ICKT(K)
      M=0
  509 DO 505 I=1,INL
      IF(IIB-II(I))504,502,504
  502 IF(JJB-JJ(I))504,503,504
  503 IF(ICCT-ICTNO(I))504,506,504
  504 IF(I-INL)505,2,505
  505 CONTINUE
  506 ISTCD(I)=1
      IF(M)508,507,508
  507 M=1
      IIIB=IIB
      IIB=JJB
      JJB=IIIB
      RTR=R(I)
      XTX=X(I)
      BIOX=BIO(I)
      GIOX=GIO(I)
      IBX=IBRTP(I)
      ICCT=ICTNO(I)
      GO TO 509
    2 WRITE(IOUT,3)
    3 FORMAT('0LINE TO BE REMOVED NOT FOUND')
      CALL EXIT
  508 CONTINUE
    5 RETURN
      END
C
C
C
      SUBROUTINE RL
C
C     REPLACES EQUIPMENT PREVIOUSLY REMOVED
C     BY SUBROUTINE DL(K).
C
      DIMENSION II(400),JJ(400),R(400),X(400),GIO(400),BIO(400)
      DIMENSION ICTNO(400),IBRTP(400),ISTCD(400)
      DIMENSION CAPTR(100)
      COMMON /LINE/ IIB,JJB,RTR,XTX,BIOX,GIOX,ICCT,IBX
      COMMON /LFC/ II,JJ,R,X,GIO,BIO,ICTNO,IBRTP,ISTCD
      COMMON /LFA/ INL
      IF(IIB.EQ.0) GO TO 520
  510 M=0
  511 INL=INL+1
      II(INL)=IIB
      JJ(INL)=JJB
      R(INL)=     RTR
      X(INL)=XTX
      GIO(INL)=GIOX
      BIO(INL)=BIOX
      ISTCD(INL)=0
      ICTNO(INL)=ICCT
      IBRTP(INL)=     IBX
```



```
      IF(M)513,512,513
  512 M=1
      IIIB=IIB
      IIB=JJB
      JJB=IIIB
      IF(IBX)515,511,515
  515 TEMP=GIOX
      GIOX=BIOX
      BIOX=TEMP
      GO TO 511
  513 CONTINUE
  520 RETURN
      END
C
C
C
      SUBROUTINE CAPCHK(X,ICNF,VC,RES,NB,N,BC)
C
C     CHECKS VOLTAGE LEVELS FOR ALLOCATION X
C     WITH LOAD FLOW ANALYSIS.
C
      DIMENSION RES(1),ICA(20),VC(1)
      DIMENSION V(100)
      DIMENSION CAPTR(100)
      INTEGER X(20)
      COMMON /SCAP/ ICA
      COMMON /SC/ CAPTR
      COMMON /VLTGE/ V
      COMMON /LFD/ ICNFG
      IOUT=6
      ICNFG=ICNF
      CALL DL(ICNFG)
      DO 10 I=1,NB
      IK=ICA(I)
   10 CAPTR(IK)=CAPTR(IK)+X(I)*BC
      CALL LF2
      CALL LF3
      DO 11 I=1,NB
      IK=ICA(I)
   11 CAPTR(IK)=CAPTR(IK)-X(I)*BC
      CALL RL
      EMAX=0.0
      J=0
      BM=0.0
      DO 20 I=1,N
      B=RES(I)-V(I)+VC(I)
      WRITE(IOUT,30) I,B
   30 FORMAT(' ',I3,' ERROR - ',F8.6)
      BZ=ABS(B)
      IF(BZ.LE.EMAX) GO TO 20
      EMAX=BZ
      BM=B
      J=I
   20 CONTINUE
      WRITE(IOUT,21) BM,J
   21 FORMAT(//'0### THE MAXIMUM ERROR OF THE LINEAR APPROXIMATION IS',
     1 1X,F8.6,' PU AT BUS NO.',I4,' ###')
      RETURN
      END
C
```

```
C
C
      SUBROUTINE LF1
C
C     PERFORMS INPUT OPERATIONS AND ARRAY
C     CONSTRUCTION FOR LOAD FLOW ANALYSIS.
C
      DIMENSION E(100),PP(100),WTGS(100),IBUST(100)
      DIMENSION F(100),QP(100),VRGS(100),IBUSS(100)
      DIMENSION H(100),S2(100),GII(100),WTLS(100),CAPTR(100)
      DIMENSION W(100),S3(100),BII(100),VRLS(100), BASV(100)
      DIMENSION R(400),RR(400),GIO(400),IPTR(400),ICTNO(400)
      DIMENSION X(400),XX(400),BIO(400),RATE(400),ISTCD(400)
      DIMENSION GIOIO(400),BIOIO(400)
      DIMENSION P(400),II(400)
      DIMENSION Q(400),JJ(400)
      DIMENSION IN1(100),IN2(100)
      DIMENSION IBRTP(400)
      DIMENSION IRTCD(400)
      DIMENSION KNAME( 30)
      DIMENSION IDENT(  4)
      COMMON /SC/ CAPTR
      COMMON /LINE/ IIB,JJB,RTR,XTX,BIOX,GIOX,ICCT,IBX
      COMMON /LFA/ INL /LFB/ INB
      COMMON /LFC/ II,JJ,R,X,GIO,BIO,ICTNO,IBRTP,ISTCD
      COMMON /LFE/ E,F,ISBN
      COMMON /LF/ PP,WTGS,IBUST,QP,VRGS,H,S2,GII,WTLS,W,S3,BII,VRLS,
     1 BASV,U,RR,IPTR,XX,RATE,GIOIO,BIOIO,P,Q,IRTCD,KNAME,IDENT,
     2 IT,ODL,ODH,ITCM,ANGLE,IN1,IN2,IBUSS
      COMMON /LFX/ CONVG
      COMMON /LFZ/ IRCTN
      COMMON /BASE/ BMVA
C     SET SYMBOLIC TAPE UNITS
      IN=5
      IOUT=6
      GO TO 33
C     SET ACCELERATION CONTANTS
    4 ODL=1.7
      ODH=1.3
C     READ CONTROL CARD
      READ(IN,6    )IRCTN,JRCTN,BMVA,CONVG,ANGLE,ITCM,IDENT(3),IDENT(4),J
     1CCN,ICCN,IDENT(1),IDENT(2)
    6 FORMAT(24X,2(I1,1X),F4.1,1X,F9.8,1X,F3.1,1X,I3,1X,A4,A4,1X,I4,1X,I
     14,1X,A4,A4)
C     CHECK NEW CASE CODE-0 IF NEW,1 IF CHANGES
      IF(IRCTN)501,500,501
C     NEW CASE,SET LINE,BUS COUNT TO ZERO
  500 INL=0
      INB=0
C     CHECK CONTROL INFORMATION,IF ZERO,SET DEFAULT VALUE
  501 IF(BMVA)8    ,7    ,8
    7 BMVA=100.0
    8 IF(ITCM)10    ,9    ,10
    9 ITCM=300
   10 IF(CONVG)12,11,12
   11 CONVG=.001
   12 IF(ANGLE)14,13,14
   13 ANGLE=45.0
   14 GO TO 39
C     READ CARD TO IDENTIFY DATA FOLLOWING
```

```
   33 READ(IN,34   )ICNTR
   34 FORMAT(79X,I1)
C     TEST CODE TO IDENTIFY DATA FOLLOWING
   35 IF(ICNTR-1) 2   ,4   ,36
   36 IF(ICNTR-3)39   ,42   ,37
   37 IF(ICNTR-5)105 ,  2 ,38
   38 IF(ICNTR-7) 178,180,180
C     READ 2 COMMENT CARDS
   39 N=1
      NO=15
      DO 41   I=1,2
      READ(IN,40   )(KNAME(K),K=N,NO)
   40 FORMAT(15A4)
      N=N+15
   41 NO=NO+15
      GO TO 33
C     READ BUS CARDS
   42 READ(IN,43   )NA1,NA2,IBN1,E2,F2,WG1,VG1,IBT1,WL1,VL1,CAP1,BV1,IA,V
     1S1,IS1,ID1,ID2,ICNTR
   43 FORMAT(2A4,3X,I3,1X,F5.4,1X,F5.4,1X,F5.1,1X,F5.1,1X,I1,1X,F4.1,1X,
     1F4.1,1X,F5.4,1X,F4.1,1X,I1,1X,F4.1,1X,I1,A4,A4,1X,I1)
C     CHECK FOR DATA CONTROL CARD
      IF(ICNTR)35   ,44   ,35
C     CHECK TO SEE IF DATA BELONGS TO CASE
   44 IF(ID1-IDENT(1)) 2   ,45   , 2
   45 IF(ID2-IDENT(2)) 2   ,48   , 2
C     CHECK BUS STATUS I.E. IN OR OUT
   48 IF(IS1-1)49   ,58   ,55
   55 IF(IS1-4)58   ,59   , 2
C     NEW BUS,SET INB=LARGEST BUS NO.
   49 INB=MAX0(INB,IBN1)
      KE=IBN1
C     CHECK FOR SWING BUS
      IF(IBT1-3)51   ,50   ,51
C     SET ISBN=BUS NO.
   50 ISBN=IBN1
C     SET SWING VOLTAGES
      E(ISBN)=E2
      F(ISBN)=F2
C     ALL NEW BUSES,LOAD ARRAYS
   51 WTGS(KE)=WG1
      VRGS(KE)=VG1
      IBUST(KE)=IBT1
      WTLS(KE)= WL1
      VRLS(KE)=VL 1
      CAPTR(KE)=CAP1
      BASV(KE)=BV1
      IN1(KE)=NA1
      IN2(KE)=NA2
      IF(IBT1-2)53   ,52   ,53
C     TYPE 2 BUS,SET SCHEDULED VOLTAGE
   52 VRLS(KE)=VS1
   53 IBUSS(KE)=IS1
      GO TO 42
C     FOR BUS OUTAGE
   58 IBUSS(IBN1)=IS1
      GO TO 42
C     FOLLOWING FOR BUS CHANGES,IF ZERO,NO CHANGE,IF 0=99,SET TO ZERO
   59 IF(IBT1-3)69   ,60   ,69
C     SWING BUS ONLY
```

```
   60 ISBN=IBN1
      IF(E2)61   ,64   ,61
   61 IF(E2-9.9999)63   ,62   ,63
   62 E(ISBN)=0.0
      GO TO 64
   63 E(ISBN)=E2
   64 IF(F2)65   ,68   ,65
   65 IF(F2-9.9999)67   ,66   ,67
   66 F(ISBN)=0.0
      GO TO 68
   67 F(ISBN)=F2
   68 CONTINUE
C     FOLLOWING FOR ALL BUS CHANGES
   69 IF(WG1)73   ,76   ,73
   73 IF(WG1-9999.9)75   ,74   ,75
   74 WTGS(IBN1)=0.0
      GO TO 76
   75 WTGS(IBN1)=WG1
   76 IF(VG1)77   ,80   ,77
   77 IF(VG1-999.99)79   ,78   ,79
   78 VRGS(IBN1)=0.0
      GO TO 80
   79 VRGS(IBN1)=VG1
   80 IF(IBT1) 81,82,81
   81 IBUST(IBN1)=IBT1
   82 IF(WL1)83   ,86   ,83
   83 IF(WL1-999.9)85   ,84   ,85
   84 WTLS(IBN1)=0.0
      GO TO 86
   85 WTLS(IBN1)=WL1
   86 IF(VL1)87   ,90   ,87
   87 IF(VL1-999.9)89   ,88   ,89
   88 VRLS(IBN1)=0.0
      GO TO 90
   89 VRLS(IBN1)=VL1
   90 IF(CAP1)91   ,94   ,91
   91 IF(CAP1-9.9999)93   ,92   ,93
   92 CAPTR(IBN1)=0.0
      GO TO 94
   93 CAPTR(IBN1)=CAP1
   94 IF(BV1)95   ,98   ,95
   95 IF(BV1-999.9)97   ,96   ,97
   96 BASV(IBN1)=0.0
      GO TO  98
   97 BASV(IBN1)=BV1
   98 IF(VS1)101 ,104 ,101
  101 IF(VS1-999.9)103 ,102 ,103
  102 VRLS(IBN1)=0.0
      GO TO 104
  103 VRLS(IBN1)=VS1
  104 GO TO 42
C     READ BRANCH CARDS
  105 READ(IN,106 )IB,JB,SR,SRE,A,B,BR,IC,ICNU,IS,IS1,IBT,IC1,IC2,ICNTR
  106 FORMAT(24X,I3,1X,I3,1X,2(F5.4,1X),2(F4.1,1X),F4.0,1X,A1,1X,I2,1X,I
     11,1X,I1,1X,I1,1X,A4,A4,1X,I1)
C     CHECK FOR DATA CONTROL CARD
      IF(ICNTR)35   ,107 ,35
C     CHECK TO SEE IF DATA BELONGS TO CASE
  107 IF(IC1-IDENT(1))  2 ,108 ,2
  108 IF(IC2-IDENT(2))2    ,110 ,2
```

```
C     CHECK BRANCH TYPE 0 IS LINE,1 IS TRANSFORMER
  110 IF(IBT)112 ,111 ,112
C     LINE,GET FORMAT BACK TO 4.3 FOR YIO
  111 A=A/100.0
      B=B/100.0
C     CHEK STATUS (NEW,REMOVE,CHANGE)
  112 IF(IS1-1)123 ,115 ,114
  114 IF(IS1-4)115 ,129 ,2
C     REMOVE FOLLOWS
C     LOOP EXECUTED TWICE FOR IIJJ,JJII
C     M=0 FIRST TIME, M=1 SECOND TIME
  115 M=0
  601 DO 121 IL=1,INL
      IF(IB-II(IL))118 ,116 ,118
  116 IF(JB-JJ(IL))118 ,117 ,118
  117 IF(ICNU-ICTNO(IL))118 ,122 ,118
  118 IF(IL-INL)121 , 2  ,121
  121 CONTINUE
  122 ISTCD(IL)=IS1
C     IF M=0,SWAP TERMINALS,M=1,DONE
      IF(M) 105,600,105
  600 ITEM=IB
      IB=JB
      JB=ITEM
      M=1
      GO TO 601
C     FOLLOWING FOR NEW BRANCH
  123 Y=0.0
C     INCREMENT BRANCH COUNT
  124 INL=INL+1
C     LOAD ARRAYS
      II(INL)=IB
      JJ(INL)=JB
      R(INL)=SR
      X(INL)=SRE
      GIO(INL)=A
      BIO(INL)=B
      RATE(INL)=BR
      IRTCD(INL)=IC
      ICTNO(INL)=ICNU
      ISTCD(INL)=IS1
      IBRTP(INL)=IBT
C     PREPARE TO SWAP TERMINALS (I.E.  II=JJ,JJ=II)
      IF(IBT)125 ,127 ,125
C     FOLLOWING FOR TRANSFORMERS ONLY Y=0.0 FIRST TIME,1.0 SECOND
  125 IF(Y-1.0)126 ,105 ,126
C     SWAP TAPS
  126 Z=A
      A=B
      B=Z
C     SET LOOP CONTROL
      Y=1.0
      GO TO 128
C     IF IS NOT=0,DONE
  127 IF(IS)105 ,128 ,105
C     SWAP TERMINALS,ALL BRANCHS
  128 I=IB
      IB=JB
      JB=I
C     SET LOOP CONTROL
```

```
      IS=1
      GO TO 124
C     END OF NEW BRANCH
C     FOLLOWING FOR BRANCH CHANGES
C     FIND BRANCH TO BE CHANGED
  129 DO 134  I=1,INL
      IF(IB-II(I))132 ,130 ,132
  130 IF(JB-JJ(I))132 ,131 ,132
  131 IF(ICNU-ICTNO(I))132 ,135 ,132
  132 IF(I-INL)134 ,133 ,134
  133 GO TO 2
  134 CONTINUE
C     START CHECKING FIELDS,IF 0,NO CHANGE,IF 99,CHANGE TO ZERO
C     IF NEITHER,SET TO VALUE
  135 ISTCD(I)=0
      IF(SR)136 ,139 ,136
  136 IF(SR-9.9999)138 ,137 ,138
  137 R(I  )=0.0
      GO TO 139
  138 R(I  )=SR
  139 IF(SRE)140 ,143 ,140
  140 IF(SRE-9.9999)142 ,141 ,142
  141 X(I  )=0.0
      GO TO 143
  142 X(I  )=SRE
  143 IF(A)144 ,147 ,144
  144 IF(A-9.999)146 ,145 ,146
  145 GIO(I  )= 0.0
      GO TO 147
  146 GIO(I  )=A
  147 IF(B)148 ,151 ,148
  148 IF(B-9.999)150 ,149 ,150
  149 BIO(I  )=0.0
      GO TO 151
  150 BIO(I  )=B
  151 IF(BR)152 ,156 ,152
  152 IF(BR-999.9)153 ,154 ,153
  153 IF(BR-9999.0)155 ,154 ,155
  154 RATE(I  )=0.0
      IRTCD(I  )=IC
      GO TO 156
  155 RATE(I  )=BR
      IRTCD(I  )=IC
  156 IF(IBT)157 ,160 ,157
  157 IF(IBT-9)159 ,158 ,159
  158 IBRTP(I  )=0
      GO TO 160
  159 IBRTP(I  )=IBT
C     CHECK FOR SYMMETRY,OR LOOP CONTROL
  160 IF(IS)105 ,161 ,105
C     IF TRANSFORMER,SWAP TAPS
  161 IF(IBRTP(I))162 ,163 ,162
  162 C=A
      A=B
      B=C
C     ALL BRANCHS,SWAP TERMINALS
  163 I=JB
      JB=IB
      IB=I
C     SET LOOP CONTROL
```

```
      IS=1
      GO TO 129
C     ERROR ROUTINE FOLLOWS
    2 WRITE(IOUT,170)
  170 FORMAT('0INPUT DATA IN ERROR')
      CALL EXIT
C     NON-DEFAULT ACCELERATION CONTANTS
  178 READ(IN,179 )ODL,ODH
  179 FORMAT(F3.2,1X,F3.2)
      GO TO 33
  180 RETURN
      END
C
C
C
      SUBROUTINE LF2
C
C     PERFORMS ITERATIVE SOLUTION FOR LOAD
C     FLOW ANALYSIS.
C
      DIMENSION E(100),PP(100),WTGS(100),IBUST(100)
      DIMENSION F(100),QP(100),VRGS(100),IBUSS(100)
      DIMENSION H(100),S2(100),GII(100),WTLS(100),CAPTR(100)
      DIMENSION W(100),S3(100),BII(100),VRLS(100), BASV(100)
      DIMENSION R(400),RR(400),GIO(400),IPTR(400),ICTNO(400)
      DIMENSION X(400),XX(400),BIO(400),RATE(400),ISTCD(400)
      DIMENSION GIOIO(400),BIOIO(400)
      DIMENSION P(400),II(400)
      DIMENSION Q(400),JJ(400)
      DIMENSION IN1(100),IN2(100)
      DIMENSION VT(100)
      DIMENSION IBRTP(400)
      DIMENSION IRTCD(400)
      DIMENSION KNAME( 30)
      DIMENSION IDENT(  4)
      COMMON /LINE/ IIB,JJB,RTR,XTX,BIOX,GIOX,ICCT,IBX
      COMMON /SC/ CAPTR
      COMMON /LFA/ INL /LFB/ INB
      COMMON /LFC/ II,JJ,R,X,GIO,BIO,ICTNO,IBRTP,ISTCD
      COMMON /LFE/ E,F,ISBN
      COMMON /LF/ PP,WTGS,IBUST,QP,VRGS,H,S2,GII,WTLS,W,S3,BII,VRLS,
     1 BASV,U,RR,IPTR,XX,RATE,GIOIO,BIOIO,P,Q,IRTCD,KNAME,IDENT,
     2 IT,ODL,ODH,ITCM,ANGLE,IN1,IN2,IBUSS
      COMMON /LFZ/ IRCTN
      COMMON /LFX/ CONVG
      COMMON /BASE/ BMVA
      COMMON /VLTGE/ VT
C     ZERO SORT POINTERS
  192 DO 164 I=1,INL
  164 IPTR(I)=0
C     SET TAPE UNIT
      IOUT=6
C     ZERO VOLTAGE CORRECTION ARRAYS
      DO 440 I=1,INB
      H(I)=0.0
  440 W(I)=0.0
C     PREPARE TO SET COUNTERS FOR SORT
C     III WILL BE POSITION IN LIST IE POINTER VALUE
C     I IS BUS NO
      III=1
      KO=INL
      DO 165 I=1,INB
      DO 165 J=1,INL
      IF(II(J)-I)165,166,165
C     CHECK TO SEE IF BRANCH IS OUT
  166 IF(ISTCD(J)-1)167,170,167
  167 JE=II(J)
C     CHECK TO SEE IF CONNECTED TO BUS OUT
      IF(IBUSS(JE)-1)168,170,168
  168 JE=JJ(J)
      IF(IBUSS(JE)-1)169,170,169
C     BRANCH CONNECTED TO BUS I,NOT OUT,SET POINTER VALUE
  169 IPTR(J)=III
      III=III+1
      GO TO 165
C     LINE OUT OR CONNECTTED TO BUS OUT,REDUCE LINE COUNT
  170 KO=KO-1
  165 CONTINUE
C     SORT ARRAYS IN POINTER ORDER,IGNORE POINTER VALUES OF 0
C     I IS POSITION IN LIST
C     J IS POINTER VALUE
      J=1
      I=1
C     LOOK FOR POINTER VALUE J,IN I POSITION IN LIST
  173 IF(IPTR(I)-J)172,171,172
C     SWAP I AND J POSITIONS
  171 IIT=II(J)
      JJT=JJ(J)
      RT=R(J)
      XT=X(J)
      GIOT=GIO(J)
      BIOT=BIO(J)
      RATET=RATE(J)
      ICTNT=ICTNO(J)
      ISTCT=ISTCD(J)
      IBRTT=IBRTP(J)
      IPTRT=IPTR(J)
      IRTCT=IRTCD(J)
      II(J)=II(I)
      JJ(J)=JJ(I)
      R(J)=R(I)
      X(J)=X(I)
      GIO(J)=GIO(I)
      BIO(J)=BIO(I)
      RATE(J)=RATE(I)
      ICTNO(J)=ICTNO(I)
      ISTCD(J)=ISTCD(I)
      IPTR(J)=IPTR(I)
      IBRTP(J)=IBRTP(I)
      IRTCD(J)=IRTCD(I)
      II(I)=IIT
      JJ(I)=JJT
      R(I)=RT
      X(I)=XT
      GIO(I)=GIOT
      BIO(I)=BIOT
      RATE(I)=RATET
      ICTNO(I)=ICTNT
      IPTR(I)=IPTRT
      ISTCD(I)=ISTCT
      IBRTP(I)=IBRTT
      IRTCD(I)=IRTCT
C     SET NEXT POINTER VALUE
      J=J+1
C     RESET TO  BEGININING OF LIST
      I=0
C     INCREMENT LIST POSITION
  172 I=I+1
C     TEST FOR END OF SEARCH
      IF(J-KO) 173,173,174
C     SET NEW LINE COUNT
  174 INL=KO
      L=1
      I=1
C     CONVERT IMPEDANCES TO ADMITTANES,STORE IMPEDENCES IN
C     TEMPORARY ARRAYS FOR LATER USE
   16 A=R(I)**2+X(I)**2
      GIOIO(I)=GIO(I)
      BIOIO(I)=BIO(I)
      RR(I)=R(I)
      XX(I)=X(I)
      R(I)=R(I)/A
      X(I)=-X(I)/A
  194 I=I+1
   17 IF(I-INL)16  ,16  ,18
C     FOLLOWING TO DIVIDE ADMITTANCES AND LOADS BY YII(TYPE 1 BUS)
C     TYPE 2 BUS TREATED DIFFERENTLY
   18 DO 29   I=1,INL
C     CHECK FOR FIRST TIME THRU
      IF(I-1)19  ,28  ,19
C     CHECK FOR SAME BUS
   19 IF(II(I)-II(I-1))24  ,20  ,24
C     CHECK BRANCH TYPE
   20 IF(IBRTP(I)-1)22  ,21  ,22
C     TRANSFORMER-GET EQUIVALENT
   21 E1=GIO (I)/BASV(J)
      K=JJ(I)
      E2=BIO (I)/BASV(K)
      A=(E2-E1)/E1
      R1=1.0/(E1*E2)
      R(I)=R(I)*R1
      X(I)=X(I)*R1
      GIO(I)=A*R(I)
      BIO(I)=A*X(I)
C     ALL BRANCHES-SUM G AND B
   22 GII(J)=GII(J)+   R(I)+GIO(I)
      BII(J)=BII(J)+   X(I)+BIO(I)
      LI=LI+1
C     CHECK FOR LAST LINE
      IF(I-INL)29  ,23  ,23
   23 L=0
C     GET P.U. MW
   24 P1=(WTGS(J)-WTLS(J))/BMVA
C     TEST FOR BUS TYPE 2-THESE TREATED DIFFERENTLY
      IF(IBUST(J)-2)25,27,25
C     TYPE 1 BUS CALCULATIONS FOLLOW
   25 Q1=(VRGS(J)-VRLS(J))/BMVA
C     DIVIDE P,Q,BRANCH Y,BY YII
      A=GII(J)**2+BII(J)**2
      PP(J)=( P1*GII(J)-Q1*BII(J))/A
      QP(J)=( Q1*GII(J)+P1*BII(J))/A
      DO 26   LO=1,LI
      K=I-LI+LO-L
      B=R(K)
      C=X(K)
      R   (K)=(+B*GII(J)+C*BII(J))/A
   26 X   (K)=( C*GII(J)-B*BII(J))/A
      GO TO  28
C     TYPE 2 BUS CALCULATIONS
   27 V=(VRLS(J)/BASV(J))**2
      PP(J)=GII(J)-P1/V
      QP(J)=V
C     INITIALIZE FOR NEW BUS
   28 J=II(I)
      LI=0
      GII(J)=0.0
      BII(J)=CAPTR(J)
C     TEST FOR LAST BUS
      IF(I-INL)20   ,29   ,29
   29 CONTINUE
C     START ITERATIVE ROUTINE
C     JO,JOT ACCELERATION CONTROL
  200 JO=0
      JOT=0
      KOUNT=0
C     STORE VALUES TEMPORARY -- THEY GET CHANGED
      TODL=ODL
      TODH=ODH
      IT=0
      N=1
      MAX=11
      SMX=3.5
C     START NEW ITERATION
  201 S=0.0
      T=0.0
      ERROR =0.0
      IF(JOT)202 ,203 ,202
  202 ALPH1=ALPHA
      BET1=BETA
  203 ALPHA=0.0
      BETA=0.0
      M=0
      DO 217  I=1,INL
      KE=II(I)
  204 IF(M)205,206,205
  205 IF(II(I)-II(I-1))208 ,206 ,208
  206 KO=JJ(I)
      M=1
C     GET CURRENT FLOW
      S=S-(E(KO)*R(I)-F(KO)*X(I))
      T=T-(E(KO)*X(I)+F(KO)*R(I))
      IF(I-INL)217 ,207 ,207
  207 KI=II(I)
      GO TO 209
  208 KI=II(I-1)
  209 IF(JO)210 ,211 ,210
C     STORE OLD CORRECTIONS FOR EXTRAPOLATION
  210 S2(KI)=H(KI)
      S3(KI)=W(KI)
C     CHECK BUS TYPE,IF 3,NO VOLTAGE CORRECTION
```

```
  211 IF(IBUST(KI)-2)213,212,1
C     VOLTAGE CORRECTION FOR TYPE 2
  212 D1=E(KI)*E(KI)+F(KI)*F(KI)
      D=D1*PP(KI)+E(KI)*S+F(KI)*T
      DP=-D
      D=QP(KI)-D1
      BIL= -PP(KI)*D+DP
      ERROR=ERROR+ABS(BIL)+ABS(D)
      H(KI)=((D*PP(KI)-DP*ODL)*F(KI)+ .5*D*T)/(E(KI)*T-F(KI)*S)
      W(KI)=((DP*ODL-D*PP(KI))-H(KI)*S)/T
      GO TO 214
C     VOLTAGE CORRECTION FOR TYPE 1
  213 P3=E(KI)*(E(KI)+S)+F(KI)*(F(KI)+T)
      DP=PP(KI)-P3
      Q3=F(KI)*S-E(KI)*T
      DQ=QP(KI)-Q3
      ERROR=ERROR+ABS(DQ)+ABS(DP)
      Z=E(KI)+E(KI)+S
      V=F(KI)+F(KI)+T
      H(KI)=(DP*S*ODH-V*DQ*ODL)/(Z*S+V*T)
      W(KI)=(H(KI)*T+DQ*ODL)/S
C     ADD VOLTAGE CORRECTION
  214 E(KI)=E(KI)+H(KI)
      F(KI)=F(KI)+W(KI)
C     FOR EXTRAPOLATION
      ALPHA=ALPHA+ABS(H(KI))
      BETA=BETA+ABS(W(KI))
      IF(JO)215 , 1  ,215
C     SUM USED IN EXTRAPOLATION
  215 S2(KI)=S2(KI)+H(KI)
      S3(KI)=S3(KI)+W(KI)
C     ZERO CURRENT SUM
    1 S=0.0
      T=0.0
C     CHECK FOR DIVERGENCE
      VOLT=E(KI)**2+F(KI)**2
      IF(VOLT-2.25) 216,242,242
  216 IF(I-INL)206 ,218 ,218
  217 CONTINUE
C     INEREMENT ITERATION COUNT
  218 IT=IT+1
C     CHECK FOR MAXIMUM ITERATION
      IF(IT-ITCM) 221,242,242
  221 S1=ERROR
C     CHECK FOR CONVERGENCE
      IF(S1-CONVG) 242,222,222
C     CHECK TO SEE IF TIME TO UNDERACCELERATE
  222 C=SMX*CONVG
      IF(S1-C)223 ,223 ,224
C     SET NEW CONTANTS
  223 SMX=0.0
      ODL=.6
      ODH=.4
      ITCM=IT+10
C     FOR EXTRAPOLATION CHECK
  224 KE=10*N+8
      IF(KE-IT)226 ,225 ,201
C     SET EXTRAPOLATION FLAG
  225 JOT=1
      GO TO 201
```

```
C     CHECK TO SEE NEXT STAGE OF EXTRAPOLATION
  226 ITTT=IT-KE
      GO TO (227,232,233),ITTT
C     EXTRAPOLATION CONSTANT
  227 JOT=0
      RO1=(ALPHA/ALPH1+BETA/BET1)
      Z=RO1*RO1
      IF(Z-3.85)228 ,229 ,229
  228 UU=Z/(4.0-Z)
      GO TO 201
  229 IF(KOUNT-2)231 ,230 ,230
C     FORCE EXTRAPOLATION
  230 KOUNT=0
      GO TO 228
  231 N=N+1
      KOUNT=KOUNT+1
      GO TO 201
  232 JO=1
      GO TO 201
C     EXTRAPOLATE
  233 DO 235  I=1,INB
      IF(I-ISBN)234 ,235 ,234
  234 E(I)=E(I)+UU*S2(I)
      F(I)=F(I)+UU*S3(I)
  235 CONTINUE
      N=N+1
      JO=0
      GO TO 201
C     RESTORE ACCELERATION,IMPEDANCES
  242 ODL=TODL
      ODH=TODH
      DO 900 I=1,INL
      R(I)=RR(I)
      GIO(I)=GIOIO(I)
      BIO(I)=BIOIO(I)
  900 X(I)=XX(I)
      DO 990 I=1,INB
      VP=E(I)**2+F(I)**2
      VP=SQRT(VP)
  990 VT(I)=VP
      RETURN
      END
C
C
C
      SUBROUTINE LF3
C
C     PERFORMS OUTPUT OPERATIONS FOR
C     LOAD FLOW ANALYSIS.
C
      DIMENSION V(100)
      DIMENSION E(100),PP(100),WTGS(100),IBUST(100)
      DIMENSION F(100),QP(100),VRGS(100),IBUSS(100)
      DIMENSION H(100),S2(100),GII(100),WTLS(100),CAPTR(100)
      DIMENSION W(100),S3(100),BII(100),VRLS(100), BASV(100)
      DIMENSION R(400),RR(400),GIO(400),IPTR(400),ICTNO(400)
      DIMENSION X(400),XX(400),BIO(400),RATE(400),ISTCD(400)
      DIMENSION GIOIO(400),BIOIO(400)
      DIMENSION P(400),II(400)
      DIMENSION Q(400),JJ(400)
      DIMENSION IN1(100),IN2(100)
      DIMENSION IBRTP(400)
      DIMENSION IRTCD(400)
      DIMENSION KNAME( 30)
      DIMENSION IDENT(  4)
      COMMON /SC/ CAPTR
      COMMON /LFA/ INL /LFB/ INB
      COMMON /LFC/ II,JJ,R,X,GIO,BIO,ICTNO,IBRTP,ISTCD
      COMMON /LFE/ E,F,ISBN
      COMMON /LINE/ IIB,JJB,RTR,XTX,BIOX,GIOX,ICCT,IBX
      COMMON /VLTGE/ V
      COMMON /LF/ PP,WTGS,IBUST,QP,VRGS,H,S2,GII,WTLS,W,S3,BII,VRLS,
     1 BASV,U,RR,IPTR,XX,RATE,GIOIO,BIOIO,P,Q,IRTCD,KNAME,IDENT,
     2 IT,ODL,ODH,ITCM,ANGLE,IN1,IN2,IBUSS
      COMMON /LFZ/ IRCTN
      COMMON /LFX/ CONVG
      COMMON /LFD/ INCFG
      COMMON /BASE/ BMVA
      DATA IACON/'A'/, IMCON/'M'/
      IOUT=6
      LI=0
      IEND=0
      AN=ANGLE
C     SET FIRST BUS FLOW TO ZERO
      P1=0.0
      Q1=0.0
C     CALCULATE BRANCH FLOWS
      DO 408  JO=1,INL
C     SET SUBSCRIPTS FOR TERMINAL BUSES
      I=II(JO)
      J=JJ(JO)
      G1=GIO(JO)
      B1=BIO(JO)
C     CALCULATE ADMITTANCES
      A=R(JO)**2+X(JO)**2
      G=R(JO)/A
      B=-X(JO)/A
C     CHECK BRANCH TYPE
      IF(IBRTP(JO)-1)407 ,406 ,407
C     EQUIVALENT CIRCUIT FOR TRANSFORMERS
  406 E1=G1/BASV(I)
      E2=B1/BASV(J)
      R1=1.0/(E1*E2)
      A=(E2-E1)/E1
      G=G*R1
      B=B*R1
      G1=A*G
      B1=A*B
C     FOR ALL BRANCHES
  407 ES=E(I)*E(I)
      FS=F(I)*F(I)
      P(JO)=(ES+FS)*(G+G1)+B*(E(I)*F(J)-E(J)*F(I))-G*(E(I)*E(J)+F(I)*F(J
     1))
      P(JO)=P(JO)*BMVA
      Q(JO)=-(ES+FS)*(B+B1)+G*(E(I)*F(J)-E(J)*F(I))+B*(E(I)*E(J)+F(I)*F(
     1J))
  408 Q(JO)=Q(JO)*BMVA
C     WRITE TITLE PAGE
      WRITE(IOUT,409 )
  409 FORMAT(1H1)
```

```
      WRITE(IOUT,410 )IDENT(1),IDENT(2)
  410 FORMAT(1H ,/////////////////////////    /,30X,55HCONSUMERS POWER C
     1OMPANY SUBTRANSMISSION LOAD FLOW STUDY,1X,2A4)
      WRITE(IOUT,411 )(KNAME(I),I=1,29)
  411 FORMAT(1H0,29A4)
      WRITE(IOUT,412 )IT
  412 FORMAT(1H0,45X,12HTHE CASE RAN,I4,11H ITERATIONS)
      WRITE(IOUT,200) CONVG
  200 FORMAT('0',43X,'CONVERGENCE CRITERION = ',F8.6)
      WRITE(IOUT,409)
C     START DOWN LINE LIST FOR BUS,BRANCH OUTPUT
      DO 441  KI=1,INL
C     CHECK BRANCH STATUS,0 FOR IN,1 FOR OUTAGE
      IF(ISTCD(KI)-1)413 ,441 ,413
C     SET SUBSCRIPT FOR NEAR BUS
  413 J=II(KI)
C     CHECK FOR NEAR BUS OUTRAGE
      IF(IBUSS(J)-1)414 ,441 ,414
C     SET REMOTE BUS SUBSCRIPT
  414 J=JJ(KI)
      IF(IBUSS(J)-2)415 ,441 ,415
  415 IF(ISTCD(KI)-2)416 ,441 ,416
C     CHECK FOR FIRST BUS
  416 IF(KI-1)417 ,418 ,417
C     CHECK FOR SAME BUS
  417 IF(II(KI)-II(KI-1))432 ,420 ,432
C     FIRST OR NEW BUS,CALCULATE AND PUT OUT BUS LINE
  418 JE=II(KI)
      PU=V(JE)
      CENT=PU*100.0
      VP=PU*BASV(JE)
      ANG=F(JE)/E(JE)
      ANG=(ATAN(ANG)*180.0)/3.14159265-AN
C     WRITE BUS LINE
      WRITE(IOUT,419) JE,IN1(JE),IN2(JE),VP,ANG,PU
  419 FORMAT('0'/I3,3X,2A4,3X,F6.2,1X,'KV',2X,F6.2,1X,'DEG',1X,F8.6,
     1 ' PU',2X,'LOSS',8X,'MVA',7X,'FLOW',4X,'AMPS',3X,'PERCENT RATING')
C     INCREMENT LINE COUNT
      LI=LI+3
C     FIND REMOTE TERMINALS
  420 DO  422  I=1,INL
      IF(II(I)-JJ(KI))422 ,421 ,422
  421 IF(JJ(I)-II(KI))422 ,423 ,422
  422 CONTINUE
C     POWER LOSS IN LINE
  423 PL=P(I)+P(KI)
      QL=Q(I)+Q(KI)
      VA=P(KI)**2+Q(KI)**2
      VA=SQRT(VA)
C     MVA FLOW
      FLOW=(VA*100.0)/CENT
      D=1.73205*VP
C     AMP FLOW
      JMPS=(1000.0*VA)/D
C     SEE IF PCT LOADING TO BE CHECKED
      IF(RATE(KI))425 ,424 ,425
C     NO,SET OUTPUT TO ZERO
  424 RAT=0.0
      GO TO 426
C     GET PCT LOADING,AFTER CHECKING UNITS
```

```
  425 IF(IRTCD(KI)-IACON) 445,444,445
  444 RAT=JMPS*100./RATE(KI)
      GO TO 426
  445 IF(IRTCD(KI)-IMCON) 424,446,424
  446 RAT=VA*1000./RATE(KI)
C     GET REMOTE BUS NUMBER
  426 JO=JJ(KI)
C     CHECK FOR LINE OR TRANSFORMER FORMAT
      IF(IBRTP(KI)-1)427 ,429 ,427
  427 WRITE(IOUT,428) JO,IN1(JO),IN2(JO),P(KI),Q(KI),PL,QL,VA,FLOW,JMPS,
     1 ISTCD(KI),RAT
  428 FORMAT(1H ,4X,I3,1X,2A4,3X,F7.1,1X,1HP,2X,F7.1,1X,1HQ,1X,F6.2,1X,2
     1HPL,1X,F6.2,1X,2HQL,2X,F6.1,4X,F6.1,3X,I4,1X,A1,3X,F5.2)
      GO TO 431
  429 WRITE(IOUT,430) JO,IN1(JO),IN2(JO),P(KI),Q(KI),PL,QL,VA,FLOW,JMPS,
     1 ISTCD(KI),RAT,GIO(KI),BIO(KI)
  430 FORMAT(1H ,4X,I3,1X,2A4,3X,F7.1,1X,1HP,2X,F7.1,1X,1HQ,1X,F6.2,1X,2
     1HPL,1X,F6.2,1X,2HQL,2X,F6.1,4X,F6.1,3X,I4,1X,A1,3X,F5.2,4X,F5.1,1H
     2-,F5.1,5H TAPS)
C     INCREMENT PAGE COUNTER
  431 LI=LI+1
C     TOTAL FLOW FOR BUS
      P1=P1+P(KI)
      Q1=Q1+Q(KI)
      GO TO 441
C     FINISH BUS INFORMATION
  432 CAP=(E(JE)**2+F(JE)**2)*CAPTR(JE)*BMVA
C     CACULATE P,Q MISMATCH
      PM=-WTGS(JE)+WTLS(JE)+P1
      QM=-VRGS(JE)+VRLS(JE)+Q1-CAP
      IF(IBUST(JE)-2)435 ,433 ,434
C     TYPE 2 BUS
  433 VRGS(JE)=VRGS(JE)+QM-VRLS(JE)
C     SET Q MISMATCH TO ZERO
      QM=0.0
      VRG=0.0
      GO TO 436
C     SWING BUS,SET REQUIRED GENERATION
  434 VRGS(JE)=VRGS(JE)+QM
      WTGS(JE)=WTGS(JE)+PM
C     SET P,Q MISMATCH TO ZERO
      PM=0.0
      QM=0.0
C     TYPE 1 BUS ONLY
C     FOLLOWING FOR ALL BUSES
  435 VRG=VRLS(JE)
  436 WRITE(IOUT,437 )WTGS(JE),VRGS(JE),CAP
  437 FORMAT(1H ,16X,F7.1,1X,2HPG,1X,F7.1,1X,2HQG,3X,F6.2,1X,3HCAP,10X,8
     1HMISMATCH)
      WRITE(IOUT,438 )WTLS(JE),VRG      ,PM,QM
  438 FORMAT(1H ,16X,F7.1,1X,2HPL,1X,F7.1,1X,2HQL,18X,F5.2,1X,1HP,9X,F5.
     12,1X,1HQ)
C     SET BUS FLOWS TO ZERO
      Q1=0.0
      P1=0.0
      IF(IEND)442 ,439 ,442
C     INCREMENT LINE COUNTER
  439 LI=LI+2
C     CHEK FOR FULL PAGE
      IF(LI-45)418 ,440 ,440
```

```
C     NEW PAGE,RESET LINE COUNT,PRINT HEADRESS
  440 LI=0
      WRITE(IOUT,409)
      WRITE(IOUT,450)(KNAME(NN),NN=1,29)
  450 FORMAT('0',29A4)
      GO TO 418
  441 CONTINUE
      IEND=1
      GO TO 432
  442 CONTINUE
  443 CONTINUE
      NSQ=INB*INB
      RETURN
      END
C
C
C
      SUBROUTINE ECON(X,XS,NB,COST)
C
C     COMPUTES COST OF A GIVEN CAPACITOR ALLOCATION.
C     IF INSTL(I) IS .TRUE. SWITCHED UNITS ALREADY
C     EXIST AT BUS ICA(I).  THE COST OF ADDITIONAL
C     UNITS THERE IS THE SAME AS THE COST OF FIXED
C     UNITS.  COST IS RETURNED IN THOUSANDS OF
C     DOLLARS.
C     NOTE:  $3,000 ADDED IN FIXED INSTALLATIONS
C            FOR LABOR COSTS.
C
      INTEGER X(1),XS(1)
      LOGICAL INSTL(20)
      COMMON /INSBK/ INSTL
      COMMON /EC/ CF,SG,BC
      COMMON /BASE/ BMVA
      COST=0.0
      QMVAR=BC*BMVA
      CFUNIT=CF*QMVAR
      DO 10 J=1,NB
      IF(X(J).EQ.0) GO TO 10
      IF(XS(J).NE.0) GO TO 11
      IF(INSTL(J)) GO TO 11
      COST=COST+CFUNIT*X(J)+SG
      GO TO 10
   11 COST=COST+CFUNIT*X(J)+3.0
   10 CONTINUE
      RETURN
      END
```