POWER SYSTEM EQUIVALENTS
OBTAINED BY APPROXIMATING THE TIE-LINE FLOWS

by

© Zeyad Naamani
B.ENG. (McGill University)

A thesis submitted to the
Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of
Master of Engineering.

Department of Electrical Engineering,
McGill University,
Montreal, Canada

February, 1981.

ABSTRACT

A new equivalencing technique for static power systems is introduced. In a two stage computation, the tie-line power flows are approximated by the first few terms of a Taylor series expansion about a known base case. The first stage involves determining the sensitivity matrices of the tie-line flows, which depend only on the network topology and the base case. The following stage employs the sensitivity matrices and the specified injections to approximately calculate the tie-line flows. Subsequently, the boundary bus injections are augmented by the approximate tie-line flows, and the retained system load flow may then be implemented.

Since the sensitivity matrices are independent of injections, it is straightforward to determine the new approximate tie-line flows corresponding to the changed injections. However, when network changes occur, it is necessary to recalculate the sensitivity matrices. A less computationally demanding alternative is to update the matrices. Two such methods are suggested.

Several test cases demonstrating the method and typical results are provided.

RESUME

Une nouvelle technique d'équivalence pour systèmes de transport d'énergie en régime stationnaire est introduite. Dans une computation à deux étapes, l'écoulement de puissance dans les lignes d'accouplement est approximé par les premiers termes de l'expansion de la série de Taylor basée sur un cas connu. La premiére étape consiste à déterminer les matrices de sensitivité de l'écoulement dans les lignes d'accouplement, qui dépendent uniquement de la topologie du système et du cas de base. La deuxième étape emploie les matrices de sensitivié et les injections spécifiées pour estimer l'écoulement dans les lignes d'accouplement. Ultérieurment, les injections aux circuits communs interfaciaux sont augmentées par la valeur approximative de l'écoulement dans les lignes d'accouplement, et l'écoulement de charge pour le système retenu peut être appliqué.

Puisque les matrices de sensitivité sont indépendantes des injections, il est facile de déterminer la nouvelle valeur approximative de l'écoulement dans les lignes d'accouplement correspondants aux nouvelles injections. Néenmoins, quand la topologie du système change, il est nécessaire de recalculer les matrices de sensitivité. Une alternative qui est quantitativement moins exigeante, est presentée pour modifier les matrices. Deux méthodes sont suggérées.

Plusieurs cas d'essais démontrant la méthode et des résultats typiques sont inclus.

ACKNOWLEDGEMENTS

The author wishes to express his appreciation to Dr. F.D. Galiana for his skillful guidance, enthusiasm and encouragement.

The author is greatly indepted to his fellow students, in particular, S.L. Low and M. Banakar for their availability and helpful suggestions, and P. Legault for his help in translating the abstract.

The grant, number A-4419, received from N.S.E.R.C is gratefully acknowledged.

The funds received from the McGill Computing Center for typing this thesis via the Script facility are gratefully acknowledged.

Last but not least, I would like to offer special thanks to my parents for their encouragement and financial support.

TABLE OF CONTENTS

TABLE OF CONTENTS 1

CHAPTER 1

INTRODUCTION 4

1.1 WHY ARE EQUIVALENTS NEEDED ? 4

1.2 PERFORMANCE OF THE EQUIVALENT 6

1.3 DISADVANTAGES OF USING AN EQUIVALENT 8

1.4 EXCHANGE OF INFORMATION BETWEEN UTILITIES . . . 9

1.5 ALTERNATIVE METHODS FOR HANDLING LARGE NETWORKS 10

1.6 OVERVIEW OF REDUCTION METHODS 10

1.7 OUTLINE OF THE THESIS 15

1.8 CONTRIBUTIONS OF THE THESIS 16

CHAPTER 2

REVIEW OF WARD,REI,LINEARIZED JACOBIAN EQUIVALENCING 18

2.1 INTRODUCTION 18

2.2 THE WARD EQUIVALENT 20

2.2.1 WARD REDUCTION 20

2.2.2 SHORTCOMINGS OF THE WARD REDUCTION 23

2.3 THE REI EQUIVALENT 25

2.3.1 FORMING THE REI 25

2.3.2 COMMENTS ON THE REI REDUCTION 30

2.4 LINEARIZED JACOBIAN METHOD 31

CHAPTER 3

APPROXIMATION FORMULAE FOR
DEPENDENT LOAD FLOW VARIABLES 40

3.1 OBJECTIVE 40

3.2 THE AC LOAD FLOW 41

3.3 REFORMULATING THE LOAD FLOW PROBLEM 44

3.4 THE APPROXIMATION FORMULAE 48

3.5 DERIVATION OF THE APPROXIMATION FORMULAE . . . 49

3.5.1 USEFUL PROPERTIES 49

3.5.2 FORMULAE DERIVATION 52

CHAPTER 4

THE PROPOSED EQUIVALENCING METHOD 60

4.1 MOTIVATION 60

4.2 APPROXIMATING THE TIE-LINE FLOWS 63

4.3 AUGMENTING THE BOUNDARY INJECTIONS 66

4.4 ILLUSTRATION 70

4.5 UPDATING THE EQUIVALENT 77

4.5.1 METHOD 1 78

4.5.2 METHOD 2 89

CHAPTER 5

TEST CASES 95

5.1 THE 5 BUS SYTEM 95

5.1.1 CLASSIFYING THE VARIOUS CASES 95

5.1.2 APPROACH 96

5.1.3 NOTATION 97

5.1.4 SAMPLE RESULTS 99

5.2 THE IEEE 30 BUS SYSTEM 115

5.3 COMMENTS ON THE NUMERICAL RESULTS 121

CHAPTER 6

CONCLUSIONS AND RECOMMENDATIONS 123

REFERENCES 127

APPENDIX A

R MATRICES 132

A.1 FORMING THE R MATRIX CORRESPONDING TO TRANSMISSION LOSSES 132

A.2 OTHER R MATRICES 134

APPENDIX B

APPROXIMATION PROGRAM 136

APPENDIX C

LOAD FLOW PROGRAM 175

CHAPTER 1

INTRODUCTION

1.1 WHY ARE EQUIVALENTS NEEDED ?

In the pre-digital computer era, power flows were obtained by modelling the power system network on an analog computer known as the "network analyzer". Needless to say preparing the model to simulate designated network configurations consumed much time, consequently, power flow analysis was performed off-line.

A stage was soon reached where power systems had expanded beyond the limited number of circuits and generator units available on the network analyzer. For the first time a need arose to represent the actual network by a smaller equivalent model. Various practices aimed at obtaining reduced models were commonly employed, of which the most prominent was to disregard portions of the network deemed insignificant to the overall behaviour. Ward [1], in his pioneer paper published in 1949, suggested a method similar to the Norton equivalent for obtaining a reduced model.

Shortly after their advent digital computers were utilized by the power industry. Digital computers offered several advantages over network analyzers ; power systems could be readily modelled and network changes could be quickly implemented. Moreover, digital computers were capable of handling larger networks than their analog predecessors; equivalencing became an obsolete art.

New developments and criteria were later introduced to power system operation and planning; economic dispatch had been integrated into load flow programs while security monitoring required the execution of several on-line load flow simulations. The time available for on-line computation did not however permit the luxury of simulating all of the desired configurations. In addition, power system networks, through continuous growth and interconnection, soon exceeded the digital computer memory capabilities. Once more, interest in developing equivalents was stimulated. The vicious circle between the demand for more powerful computers and the ever increasing expectations of on and off-line power system analysis methods has no end in sight today.

1.2 PERFORMANCE OF THE EQUIVALENT

The need for increasing power system reliability and for minimizing operating costs has dictated the organization of power pools, i.e, the interconnection of networks owned and operated by independent utilities. Typically, in order to run a load flow, information about the entire system should be available. However a particular utility of the power pool is primarily interested in its own system. Moreover, lack of external information, insufficient computer memory and time limitations preclude it from using the interconnected system model. Accordingly the utility is induced to develop an equivalent that isolates its own area from the rest of the system and provides a faithful representation of neighbouring systems.

The jargon used in power system equivalencing research is not mystifying and does not warrant a detailed elaboration. In order to ensure clarity, it will suffice to establish at the outset certain basic concepts and to point out commonly used synonyms. The actual power system consists of a mesh of interconnected networks that are owned and operated by various utilities. Terms used interchangeably to refer to this area are :total, full, complete, and actual system. The objective of equivalencing is to single out a particular area in the power system and retain it without introducing any modifications. Synonyms used to refer to this area are : internal, study, and retained system. The remaining network, which lies outside the area of interest and for which it is desired to find an equivalent, belongs to the external (eliminated) network. Normally, the internal and external systems are interconnected only via a few tie-lines, and the internal system buses from which the tie-lines emanate are known as boundary buses. When the internal system is adjoined to the extenal network equivalent, the reduced system model which will be used for load flow studies, is obtained.

In steady state security analysis the equivalent is used to simulate the system conditions within the area of interest while retaining the accuracy of the results within acceptable limits, when;

(1) Local disturbances such as equipment outages are present in the area.

(2) Changes in operating conditions such as load and generation levels occur throughout the system.

Other desirable properties in an equivalent are :

(1) It should contain as small a number of buses as possible.

(2) The equivalent should have a readily identifiable physical relationship to the generation and load make

up of the original system.

(3) The equivalent should be usable for interchange studies.

(4) It should be possible to adjust the equivalent so that it would be valid over a wide range of operating conditions, with little knowledge about the external system conditions.

1.3 DISADVANTAGES OF USING AN EQUIVALENT

1. An equivalent network is never exactly interchangeable with the original network. The best it can do is simulate exactly the ´base case´ used to derive the equivalent.

2. Local disturbances in the area are bound to affect the neighbouring system; this may cause some of the components in the neighbouring system to operate at their thermal or stability limits. In the equivalent, the individuality of the neighbouring systems is lost, it is no longer possible to detect such anomalous events.

3. The operating conditions of the external system are represented at the boundary buses of the equivalent,

therefore changes in boundary bus injections become difficult to trace back to the external injections.

1.4 EXCHANGE OF INFORMATION BETWEEN UTILITIES

At the outset the power industry consisted of independent utilities, each responsible for delivering power to its own region. In order to enhance reliability and minimize operating costs utilities linked their networks via a few tie-lines. Power flow exchanges were worked out in advance, there was no pressing need for operational information exchange among the pool members. Recently the trend has been to increase inter-pool data links. The IEEE committee report [19] cites that with the advanced control methods on the horizon, it might be necessary to telemeter almost as much external information as is required internally. Every pool member´s computer would have access to a data base consisting of relevant pool operating conditions that is regularly updated by the individual utilities. Among other information, such a data base would contain the status of large generating units and essential branches, it is forseen that a 30 second interval scan of these data is adequate.

1.5 ALTERNATIVE METHODS FOR HANDLING LARGE NETWORKS

Prior to presenting the historical overview, it should be mentioned that equivalencing is not the sole option available for solving networks that surpass computer memory. One feasible alternative is Network Partitioning or Diakoptics [29], where the power system is subdivided into several interconnected blocks, subsequently each block is solved separately. This method requires full knowledge of the network and suffers from the inherent disadvantage of a relatively slow solution time. Decoupling [30] is another viable alternative which exploits the weak coupling that exists between the real and reactive powers in the load flow equations. The method requires complete knowledge of the network, however in contrast to Diakoptics the solution time is relatively fast.

1.6 OVERVIEW OF REDUCTION METHODS

In his pioneer work, Ward [1] derived an equivalent that required knowledge of a base case load flow solution (denoted hereafter as the ´base case´). The base case external generations and loads are converted to either constant current sources or to constant admittance quantities. This transformation permits the external system to be expressed in terms of a linear expression relating the voltages and currents. Subsequently, Norton reduction eliminates all external system buses, and introduces new boundary bus interconnections and new boundary bus equivalent current injections which may be converted to equivalent power injections using base case values.

Duran and Arvanitidis [2] systematized the development of an equivalent. Three phases were recommended:

(1) Design phase :- consists of determining the buffer system, weak links and controlling buses.

(2) Reduction phase :- two reduction methods were proposed

a- the Norton equivalent

b- the incremental model

The authors found that the performance of method (a) was superior.

(3) Operational phase :- concerned with adjusting the real power injections that represent loads and generators at eliminated buses, and the real power injections at unobservable internal buses.

Paulsson [3] presented two methods with boundary buses being of the PV type.

(1) Norton equivalent ; the author concluded that holding the real power equivalent injections constant gives good results, whereas holding the reactive equivalent

injections constant renders poor results.

(2) This method involves finding a real power linear equivalent for the external system as seen from its terminals using a dc model. Then the real power flowing into the internal system from the tie-lines is found using an ac load flow. By equating the two real powers a set of equations is obtained which should be solved simultaneously with the internal system load flow equations.

Debs [4,6,7] and Contaxis [6,7] endeavoured to derive the equivalent, off-line, by monitoring scheduled or forced internal system outages together with state estimator data.

Dopazo, Dwarakanath, Li, Sasson [5] proposed the following for deriving an equivalent on-line:

(1) Observe boundary conditions through time using a state estimator.

(2) Model the external system by two components ;

a- Modelling the reaction of the external system by observing the variations in time of the boundary conditions. The method makes use of the p-δ portion of the load flow equations only, and uses the Kalman filter approach for recursive system identification.

b- Modelling the operating level; present conditions at the boundaries are matched by adjusting the operating level component of the model,which is interpreted as an additional load at each boundary bus.

Alvarado and Elkonyaly [8] linearized the external system about a known base case and reduced the effect of the external system to the boundary buses. At each iteration of the Newton-Raphson load flow, both the mismatch equations and the Jacobian for the retained network can be found by first completely ignoring the external network. The mismatches at the boundary buses are then corrected by an additional amount plus an amount linearly proportional to the deviation from base case conditions. The Jacobian terms corresponding to the boundary nodes are also corrected by a constant amount.
In reference [9] the authors extended their ideas to a decoupled load flow model.

Dy Liacco, Savulescu, Ramarao [12] proposed an approach that consists of deriving a topological equivalent using the REI method. A calibrating network with an arbitrary injection for on-line adjustment is subsequently adjoined to the equivalent.

Monticelli, Deckmann, Garcia, Stott [13] envisaged a simple extension to the Ward equivalent. The authors asserted that the inaccuracy of the Ward reduction is attributed to the dilemma concerning the designation of boundary buses as PV or PQ. The proposed solution was to designate all boundary buses as they actually are and adjoin,via a fictitious branch, to each PQ boundary bus a new fictitious PV bus m with Pm = 0 and Vm = base case voltage of the PQ bus.

Dopazo, Irissari, Sasson [14] suggested the following procedure for deriving an on-line equivalent :

1- Obtain a base case load flow.
2- Calculate the REI equivalent for each external area. Every equivalent consists of two nodes, one for area generation and one for area load.
3- Determine the mismatches between real time boundary conditions and those given by the equivalent. Adjust the REI node voltages to minimize the unbalances, this involves solving a linear least squares problem.
4- Adjust the equivalent transmission network parameters to further minimize boundary unbalances. Employ the Kalman filter for system identification.

Housos, Irisarri, Porter, Sasson [16] examined and appraised various Ward and REI equivalencing techniques.

They concluded that all equivalents that provide reactive support perform satisfactorily.

Deckmann, Pizzolante, Monticelli, Stott, Alsac [18] reviewed various load flow equivalencing methods, with particular emphasis on Ward, REI and Linearized methods. The paper provides a valuable insight into the principles of each method reviewed along with suggestions for improving its performance. In reference [17] the authors presented the numerical results obtained by testing the various equivalencing methods.

1.7 OUTLINE OF THE THESIS

Chapter 2 will review in detail the Ward reduction, the REI equivalencing and the Linearized Jacobian methods.

Chapter 3 will commence with a brief review of the ac load flow and will subsequently reformulate it using concise vector and matrix notation. Following that, the approximation formulae for dependent load flow variables will be presented along with their derivation.

Chapter 4 will present the motivation underlying the proposed equivalencing method, the procedure for obtaining the equivalent will be described in detail and an illustrative example will be provided. Two methods for updating the approximation will be presented.

Chapter 5 will be devoted to displaying typical results obtained while working with a 5 bus system and with the IEEE 30 bus system. The latter results will be compared to the corresponding results obtained by E. Elkonyaly [10].

Appendix A will demonstrate how the $\underline{R}$ matrices, that are needed in the approximation formulae, can be determined.

Appendix B will include the approximation program for determining dependent load flow variables, and Appendix C will include the load flow program. Both of these programs were used extensively in this research.

1.8 CONTRIBUTIONS OF THE THESIS

Approximations (linear or quadratic) to the tie-line flows are obtained by a taylor series expansion about a known base case. The reduced system comprises of the internal system, with the boundary bus injections augmented by the approximate tie-line flows. This approach does not require the modification of available load flow programs, and does not introduce any new lines or buses to the internal system. The linear approximation method belongs to the 'Linearized' methods [18], and its performance seems to be very good.

chapter 2

REVIEW OF WARD , REI , LINEARIZED JACOBIAN EQUIVALENCING

2.1 INTRODUCTION

Even though Ward's method is the oldest of equivalencing techniques, various modifications of it spur up regularly in recent literature. The main concept underlying the Ward approach is to transform the base case nonlinear external power-voltage equations into a linear current-voltage expression that is susceptible to Norton reduction. When applied, Norton reduction eliminates all external system buses and models the effect of the external network by a new set of boundary interconnections and equivalent boundary injections. Ward's reduction has been thoroughly examined in the literature, section 2.2.2 will elaborate upon its shortcomings.

In the early 1960's P. Dimo, of Romania, introduced the REI (for; Radial, Equivalent, Independent) equivalencing method. Researchers in the west, however, remained oblivious to the method until an english translation of

Dimo's book [28] was made available in 1975, since then several mutations of the REI technique have surfaced. One of the main reasons behind the interest in this method is due to the fact that the REI overcomes several of the failings to which Ward's reduction is prone. Briefly, the objective of the REI is to replace a set of external active nodes (i.e generation and load buses) by one or more fictitious active buses, thus rendering the external system buses passive and susceptible to Norton reduction. The equivalent obtained after Norton reduction has been applied comprises of the internal system with new boundary interconnections and which is adjoined to the equivalent fictitious buses.

Finally, this chapter will review the Linearized Jacobian method [8], which puts forward ideas analogous to the ones presented in this thesis. By linearizing the external system about a known base case, a set of equations relating the boundary tie-line flows to the internal system voltages are obtained. Translated into the internal system load flow problem, this specifies that at every iteration of the Newton-Raphson load flow, both the Jacobian and the boundary mismatches should be updated.

Each of the remaining three sections in this chapter will be devoted to reviewing in detail one of the above equivalencing methods.

2.2 THE WARD EQUIVALENT

2.2.1 WARD REDUCTION

Notation

$\underline{I}$ complex current injections vector.

$\underline{S}$ complex power injections vector.

$\underline{P}$ real power injections vector.

$\underline{Q}$ reactive power injections vector.

$\underline{Y}$ complex admittance matrix.

$*$ denotes the complex conjugate.

_ denotes vectors and matrices.

The power system is characterized by the linear current-voltage relationship [27] :

$$\underline{I} = \underline{Y}\ \underline{V} \tag{2.1}$$

At each bus, however, the power injections rather than the current injections are specified in practice. The ith element of $\underline{I}$ is :

$$I_i = (S_i / V_i)^* \tag{2.2}$$

Replacing (2.2) in (2.1) yields :

$$P_i - j\,Q_i = V_i^* \sum_{k=1}^{N} Y_{ik} V_k \qquad (2.3)$$

When (2.3) is written for all buses i (i=1,...,N) a set of non-linear equations relating the power injections to the bus voltages is obtained which is better known as the ´load flow equations´.

The first step towards forming the Ward equivalent is to transform the external system load flow equations of (2.3) into the corresponding form of (2.1) using a known base case load flow. Either one of two classical methods may be employed : The Ward Injection method, where all external injections are converted to constant current sources via (2.2). The Ward Admittance method, where all external injections are transformed to shunt admittances using (2.4).

$$Y_i = S_i^* / |V_i|^2 \qquad (2.4)$$

The Ward Admittance method is not a very attractive alternative [18], for it is not always appropriate to model loads by shunt admittances. Moreover, it is certainly an unreliable method for representing the Q-response at a PV bus, whose arbitrary base case power decides the value of the shunt. Furthermore, as we shall see in the following subsection, this method cannot avoid including the external shunts in the admittance matrix.

In order to determine the effect of the external system reduction upon the internal system, the base case internal power injections will also be transformed to current and admittance quantities. Thus it is possible to write the base case non-linear load flow equations in terms of a linear current-voltage relationship as in (2.1).

Partition the system admittance matrix into external, boundary and study matrices :

$$\begin{bmatrix} \underline{I}e \\ \underline{I}b \\ \underline{I}s \end{bmatrix} = \begin{bmatrix} \underline{Y}ee & \underline{Y}eb & 0 \\ \underline{Y}be & \underline{Y}bb & \underline{Y}bs \\ 0 & \underline{Y}sb & \underline{Y}ss \end{bmatrix} \begin{bmatrix} \underline{V}e \\ \underline{V}b \\ \underline{V}s \end{bmatrix} \qquad (2.5)$$

Eliminate the unknown vector $\underline{V}e$ to obtain :

$$\begin{bmatrix} \underline{I}b\ eq \\ \underline{I}s \end{bmatrix} = \begin{bmatrix} \underline{Y}bb\ eq & \underline{Y}bs \\ \underline{Y}sb & \underline{Y}ss \end{bmatrix} \begin{bmatrix} \underline{V}b \\ \underline{V}s \end{bmatrix} \qquad (2.6)$$

where

$$\underline{I}b\ eq = \underline{I}b - \underline{Y}be\ \underline{Y}ee^{-1}\ \underline{I}e \qquad (2.7)$$

$$\underline{Y}bb_{eq} = \underline{Y}bb - \underline{Y}be \; \underline{Y}ee^{-1} \; \underline{Y}eb \qquad (2.8)$$

Reduction leaves both of the internal admittance matrix ´$\underline{Y}ss$´ and the internal injections unchanged. New boundary bus interconnections and shunts emerge (2.8), and boundary current injections are modified (2.7). The equivalent boundary current injections may then be converted to power injections by using the base case load flow. Subsequently the reduced system consists of the internal system with additional boundary bus interconnections and shunts, and the equivalent boundary bus injections. Studies using the equivalent are performed by altering the internal injections or the internal network configuration.

2.2.2 SHORTCOMINGS OF THE WARD REDUCTION

Recalling that reduction was possible only after transforming the base case external generations and loads into current sources and admittances, one would expect that a transformation about a state different from the base case would yield disparate $\underline{I}b_{eq}$ and $\underline{Y}bb_{eq}$. Accordingly, one may infer that the equivalent will be an exact model only under base case conditions.

External network line charging and reactive compensation shunts require special attention. If the shunts were included in the admittance matrix of (2.5) then the equivalent boundary shunts, obtained after reduction, would acquire extremely large values. As a result, a small change in the boundary bus voltage magnitude would trigger a large consumption of reactive power; a situation which is unrealistic. To thwart such an occurrence when working with the Ward Injection method, it is sufficient to convert all the external shunts into additional bus injections before elimination. This transformation will ensure that the resulting series equivalent network will have normal X/R ratios. Unfortunately, the prospects are not so bright for the Ward Admittance method where the shunt admittances are already very large prior to reduction, with low X/R ratios. It is not unusual to end up with extremely large shunts in the reduced network. Unlike the Ward Injection method, this unpleasant situation cannot be circumvented.

In general it is more difficult to solve power flow problems after reduction is carried out. This may be attributed to the elimination of critical PV buses, the great diversity in magnitudes of the distributed injections at boundary nodes and the abnormal values of the modified admittance matrix elements. Thus, even though a particular equivalent problem is known to have a solution, the load flow might fail to converge to it with acceptable accuracy, if it converges at all!

The dilemma of classifying boundary buses as PV or PQ has received much attention [3,13]. Reference [16] concluded that the major problem with the Ward equivalent is that it does not allow for reactive power support in the equivalenced area. This can be explained by noticing that although the real power is always specified for every bus except the slack bus, the reactive power is not specified at every bus and may vary. If the equivalent assumes that the reactive power at regulated PV buses will remain at its base case value even under outages, the results are usually unacceptable. However, the results are enhanced for a Ward equivalent with a carefully selected buffer zone.

2.3 THE REI EQUIVALENT

2.3.1 FORMING THE REI

The REI approach [28] overcomes some of the disadvantages of the Ward reduction method. The idea is to replace a set of generation and/or load buses (active buses) by one or more fictitious buses connected through a lossless fictitious network to the group of active nodes which it is to replace. The equivalent power injection at the new fictitious bus is made equal to the algebraic sum of the power injections at the buses being replaced. Having done this, all buses in the external system become passive, the external network may then be reduced by Norton reduction. Consequently, the reduced system consists of the internal system having new boundary bus interconnections and shunts due to the reduction process, and of the new fictitious equivalent buses.

Fig 1 represents the actual network classified into an internal system and an external system (having distinct active and passive nodes). There are N active buses in the external system, and the power injection at any active bus i is specified as S_i (i=1,...,N). The objective is to replace all the active external buses by a fictitious active bus R connected to the buses which it is to replace by the yet undetermined admittances Y_R and Y_i (fig 2).

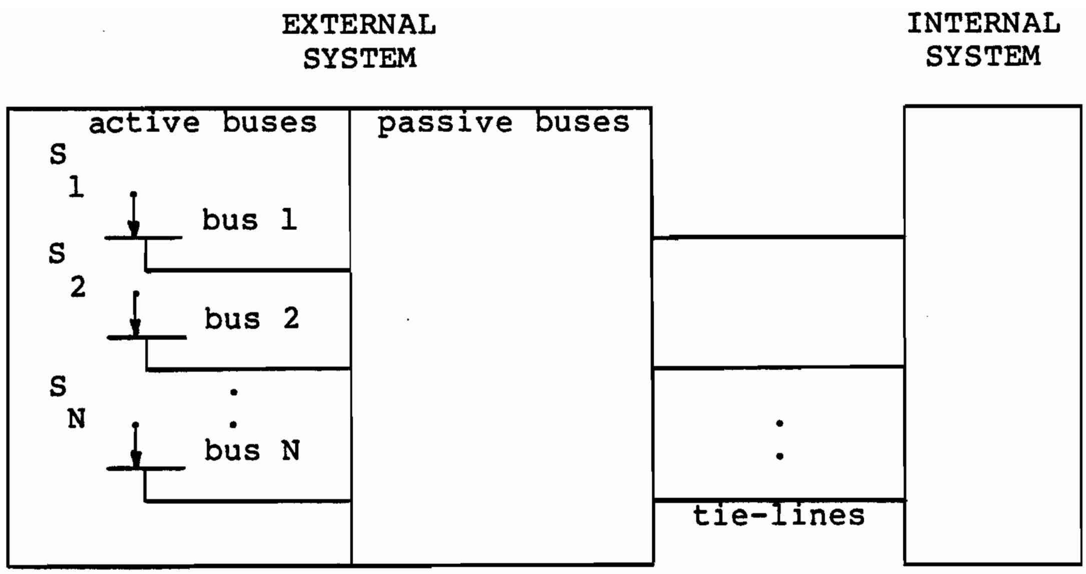

EXTERNAL SYSTEM
INTERNAL SYSTEM
active buses
passive buses
S 1
bus 1
S 2
bus 2
S N
bus N
tie-lines

FIG 1

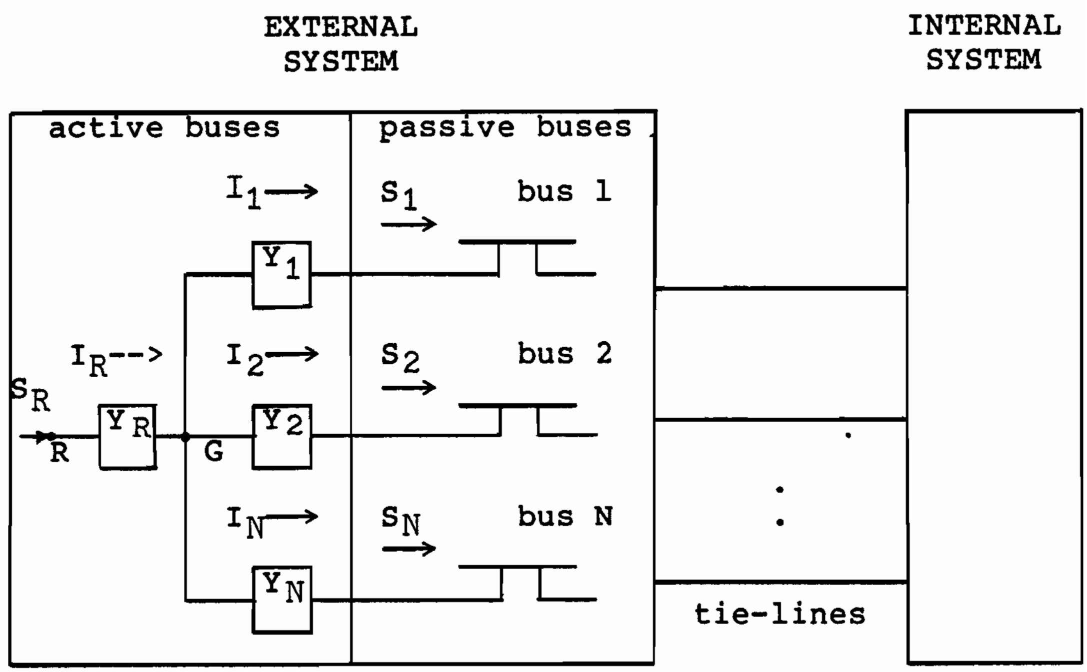

EXTERNAL SYSTEM
INTERNAL SYSTEM
active buses
passive buses
I1
Y1
S1
bus 1
IR-->
SR
YR
R
G
I2
Y2
S2
bus 2
IN
YN
SN
bus N
tie-lines

FIG 2

Bus R is the new equivalent fictitious bus, whereas bus G is a passive bus with an arbitrarily assigned voltage V_G.

The only constraints which must be satisfied are :

$$S_R = \sum_{i=1}^{N} S_i \tag{2.9}$$

and

$$I_R = \sum_{i=1}^{N} I_i \tag{2.10}$$

Accordingly voltage V_R must then be:

$$V_R = S_R \,/\, I_R^* \tag{2.11}$$

The current flowing through admittance Y_i is given by :

$$I_i = S_i^* \,/\, V_i^* \tag{2.12}$$

If $V_G = 0$ the branch admittances are

$$Y_i = - I_i \,/\, V_i = - S_i^* \,/\, |V_i|^2 \tag{2.13}$$

$$Y_R = I_R \,/\, V_R = S_R^* \,/\, |V_R|^2 \tag{2.14}$$

If $V_G \neq 0$ the branch admittances are

$$Y_i = I_i / (V_G - V_i) \tag{2.15}$$

$$Y_R = I_R / (V_R - V_G) \tag{2.16}$$

Substituting (2.8) into (2.15) , we get

$$Y_i = S_i^* / (V_i^* (V_G - V_i)) \tag{2.17}$$

$$Y_i = S_i^* / (V_i^* V_G - |V_i|^2) \tag{2.18}$$

If V_G is set equal to zero, Y_i (2.13) is sensitive only to the voltage magnitude variation . This selection is attractive in power systems because voltage magnitudes undergo slight variations as network conditions change . However if $V_G \neq 0$, Y_i (2.18) will also depend on the voltage angles which tend to vary significantly as network conditions change.

Since node G is passive it can be eliminated, however from the standpoint of sparsity it may not be advantageous to eliminate it. On the other hand, if node G is retained it may adversely affect some of the load flow algorithms that count on all network voltages to be near their nominal values.

2.3.2 COMMENTS ON THE REI

External line charging and reactive compensation shunts create the same problems for the REI as they did for the Ward Injection. These difficulties may be circumvented, as was done before, by converting the shunts to additional bus injections prior to reduction.

Reference [18] gives an enlightening discussion of accuracy considerations for the REI technique. One point of concern is the portion of power flowing from bus R to the original active buses at different operating conditions. Usually the branch admittances Y_i are small, relative to those in the rest of the system, and thus S_R will tend to be distributed in the same proportion; this behaviour is suitable for a grouping of PQ buses. Deviations from the constant proportionality condition increase as the injections become larger, thus yielding larger branch admittances (2.13). The authors also argue that the REI has a built in tendency to be ill-conditioned, especially as far as the decoupled load flow is concerned. The unusual fictitious REI branches lead to series admittances in the reduced network that may acquire unusual values. Concerning the application of the REI it is noted that, like the Ward Admittance but unlike the Ward Injection, the network admittances always retain information from the base case conditions. This information cannot be adapted to changes in external states. This limitation does not seem to be a serious drawback, for the accuracy of the boundary matched REI equivalent has been reported to be satisfactory.

Tinney and Powell [11] offer several suggestions regarding the application of the REI and sparsity programming. Dy liacco, Savulescu, Ramarao [12] use the REI method to develop their X-REI model that includes on-line calibration. Housos, Irissari, Porter, Sasson [16] investigate several methods of forming the REI. Wu and Narasimhamurthi [15] give a detailed analysis of the REI and the necessary (but not sufficient) conditions for the REI equivalent to be incrementally accurate about the initial base case.

2.4 LINEARIZED JACOBIAN METHOD [8,9,10]

Fig 3 shows a typical power system subdivided into internal and external networks, with boundary buses as well as all of the boundary bus shunts and injections belonging to the internal system. The tie-lines, which emanate from boundary buses, interconnect the internal and external networks.

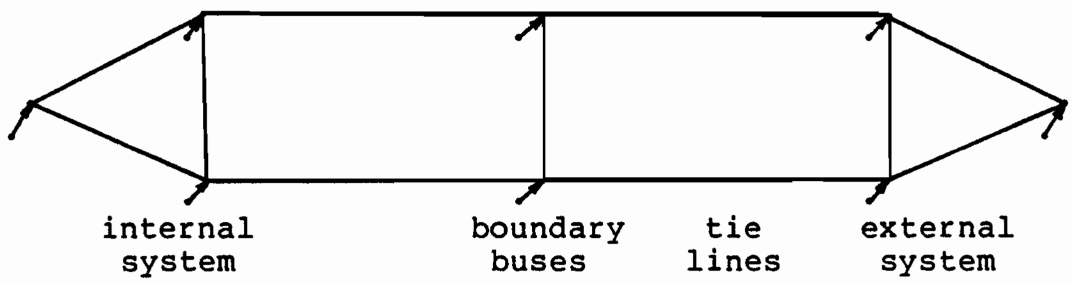


FIG 3 A Typical Power System

Subnetwork B, fig 4, consists of the internal system with boundary injections supplemented by the base case tie-line flows.

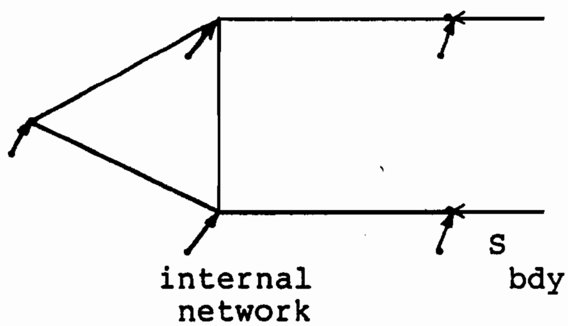


FIG 4 Subnetwork B

Subnetwork A, fig 5, consists of the external system and the tie-lines which are excited at the detached end by the base case tie-line flows.

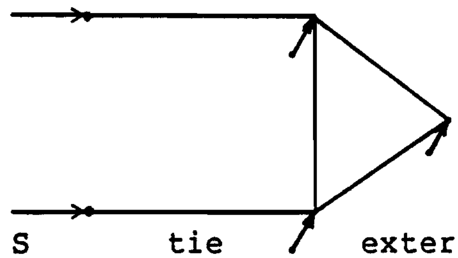


FIG 5 Subnetwork A

The separate analysis, at base case conditions, of of subnetworks A and B would yield identical results as the base case of the complete network.

Let $\underline{h}$ denote the set of mismatch equations for the external network. These are a function of the eliminated node voltages and angles (denoted by $\underline{x}_1$) and also a function of some of the retained node voltages and angles (denoted by $\underline{x}_2$).

$$h\ (\underline{x}_1, \underline{x}_2) = 0 \qquad (2.19)$$

Linearizing about the base case solution; $\underline{x}_1^o$ and $\underline{x}_2^o$

$$\underline{J}_1 \left(\underline{x}_1 - \underline{x}_1^{o}\right) + \underline{J}_2 \left(\underline{x}_2 - \underline{x}_2^{o}\right) = 0 \tag{2.20}$$

where

$$\underline{J}_1 = \left. \frac{\partial \, \underline{h} \, (\underline{x}_1, \underline{x}_2)}{\partial \, \underline{x}_1} \right|_{\underline{x}_1^{o}, \, \underline{x}_2^{o}} \tag{2.21}$$

$$\underline{J}_2 = \left. \frac{\partial \, \underline{h} \, (\underline{x}_1, \underline{x}_2)}{\partial \, \underline{x}_2} \right|_{\underline{x}_1^{o}, \, \underline{x}_2^{o}} \tag{2.22}$$

Rewriting (2.20) so as to express the unknown external voltages in terms of the internal voltages,

$$\Longrightarrow \left(\underline{x}_1 - \underline{x}_1^{o} \right) = - \underline{J}_1^{-1} \, \underline{J}_2 \left(\underline{x}_2 - \underline{x}_2^{o} \right) \tag{2.23}$$

Let $\underline{h}'$ denote the set of mismatches at the detached tie-line ends in subnetwork A, and let $\underline{S}_{bdy}$ denote the injection from subnetwork A into subnetwork B.

$$\underline{S}_{bdy} = \underline{h}' \, (\underline{x}_1, \underline{x}_2) \tag{2.24}$$

Linearizing (2.24) about the base case,

$$\underline{S}_{bdy} = \underline{S}_{bdy}^{o} + \underline{J}_3 \left(\underline{x}_1 - \underline{x}_1^{o}\right) + \underline{J}_4 \left(\underline{x}_2 - \underline{x}_2^{o}\right) \tag{2.25}$$

where

$$\underline{J}_3 = \left. \frac{\partial \, \underline{h}' (\underline{x}_1, \underline{x}_2)}{\partial \, \underline{x}_1} \right|_{\underline{x}_1^o , \, \underline{x}_2^o} \qquad (2.26)$$

$$\underline{J}_4' = \left. \frac{\partial \, \underline{h}' (\underline{x}_1, \underline{x}_2)}{\partial \, \underline{x}_2} \right|_{\underline{x}_1^o , \, \underline{x}_2^o} \qquad (2.27)$$

Therefore the Jacobian for subnetwork A is

$$\underline{J}_A = \begin{bmatrix} \underline{J}_1 & \underline{J}_2 \\ \underline{J}_3 & \underline{J}_4' \end{bmatrix} \qquad (2.28)$$

Using (2.23) to eliminate the unknown voltage vector $(\underline{x}_1 - \underline{x}_1^o)$ in (2.25)

$$\underline{S}_{bdy} = \underline{S}_{bdy}^o + (\underline{J}_4' - \underline{J}_3 \, \underline{J}_1^{-1} \, \underline{J}_2) \, (\underline{x}_2 - \underline{x}_2^o) \qquad (2.29)$$

Define $\underline{J}_{cor} = \underline{J}_4' - \underline{J}_3 \, \underline{J}_1^{-1} \, \underline{J}_2$ (2.30)

Equation (2.29) expresses the tie-line flows at the boundary buses as a function of the internal system voltages only and thus permits solving subsystem B, independently of subsystem A, at conditions different from base case.

Let $\underline{g}(\underline{x}_2)$ be the mismatch equations of the internal system, hence the mismatch equations for subnetwork B, $\underline{h}_B(\underline{x}_2)$, are given by :

$$\underline{S}_{bdy} + \underline{g}(\underline{x}_2) = 0 \tag{2.31}$$

substituting for $\underline{S}_{bdy}$;

$$\Longrightarrow \quad \underline{S}^{o}_{bdy} + \underline{J}_{cor} (\underline{x}_2 - \underline{x}^{o}_2) + \underline{g}(\underline{x}_2) = 0 \tag{2.32}$$

or equivalently,

$$\underline{h}_B (\underline{x}_2) = 0 \tag{2.33}$$

which can be solved by Newton's method ; the Kth iteration being :

$$\underline{h}_B (\underline{x}^{k}_2) = \underline{h}_B (\underline{x}^{k-1}_2) + \underline{J}^{k}_B (\underline{x}^{k}_2 - \underline{x}^{k-1}_2) = 0 \tag{2.34}$$

where

$$\underline{J}^{k}_B = \left. \frac{\partial \, \underline{h}_B (\underline{x}_2)}{\partial \, \underline{x}_2} \right|_{\underline{x}^{k-1}_2} \tag{2.35}$$

Substituting (2.32) into (2.34) ;

$$S^{o}_{bdy} + J_{cor}(x_2^{k-1} - x_2^{o}) + g(x_2^{k-1}) + J_B^{k}(x_2^{k} - x_2^{k-1}) = 0 \tag{2.36}$$

with

$$\underline{J}_B^{k} = \underline{J}_{cor} + \left. \frac{\partial\, \underline{g}(\underline{x}_2)}{\partial\, \underline{x}_2} \right|_{\underline{x}_2^{k-1}} \tag{2.37}$$

The second term of (2.37) is the Jacobian of the isolated retained network.

<u>SOLUTION ALGORITHM</u>

1. Run the base case load flow .
2. Compute Jacobians $\underline{J}_A$, $\underline{J}_{cor}$.
3. a- Guess $\underline{x}_2$.

   b- Calculate the mismatches $\underline{h}_B(\underline{x}_2)$, and the Jacobian of the retained system .

   c- Correct the mismatches at the boundary nodes by

   $\underline{J}_{cor}(\underline{x}_2^{k-1} - \underline{x}_2^{o})$.

   d- Correct the Jacobian terms corresponding to the boundary nodes by adding the terms of $\underline{J}_{cor}$ to it to obtain $\underline{J}_B$.

e- Solve $\underline{J}_B \; \Delta\underline{x}_2 = \underline{h}_B(\underline{x}_2)$ for $\Delta\underline{x}_2$.

f- Let $\underline{x}_2^{k+1} = \underline{x}_2^{k} - \Delta\underline{x}_2$.

g- Go to (b) .

The authors [8,9,10] suggest a method for simulating the effect of external network changes after reduction. By repeating the above elimination process with equation (2.20) rewritten as :

$$\underline{J}_1 (\underline{x}_1 - \underline{x}_1^{o}) + J_2 (\underline{x}_2 - \underline{x}_2^{o}) = \Delta\underline{h} \qquad (2.38)$$

The boundary injections can be expressed entirely in terms of the retained system variables :

$$\underline{S}_{bdy} = \underline{S}_{bdy}^{o} + \underline{J}_{cor} (\underline{x}_2 - \underline{x}_2^{o}) + \underline{J}_3 \underline{J}_1^{-1} \Delta\underline{h} \qquad (2.39)$$

Once more it is possible to incorporate this equation into the Newton-Raphson iterations.

In references [9,10] the authors extended the model to a form more compatible with the decoupled load flow. The Linearized Jacobian is a relatively recent contribution and has yet to receive its fair share of testing. Numerical results given by the authors [8,9,10] and in reference [17] indicate the high performance of the Linearized Jacobian as compared to other equivalencing methods. The greatest disadvantage of the method,as far as implementation is concerned, is the necessity to modify the Jacobian of the retained system. This implies that normal load flow programs cannot be directly applied to the retained system without some modification.

chapter 3

APPROXIMATION FORMULAE FOR DEPENDENT LOAD FLOW VARIABLES

3.1 OBJECTIVE

Explicit approximation formulae [23] based on the Taylor series expansion are derived relating an arbitrary dependent load flow variable, y, to the independent injections $\underline{z}$ of a general load flow problem. Two approximations will be considered :

1. The linear approximation

$$y = \underline{B}^T \underline{z} \tag{3.1}$$

2. The quadratic approximation

$$y = \underline{B}^T \underline{z} + \underline{z}^T \underline{C} \underline{z} \tag{3.2}$$

3.2 THE AC LOAD FLOW

In steady state analysis the power system is treated as a balanced three phase system and may be represented by a single phase positive sequence network. Under such conditions the static behaviour can be described in terms of a set of non-linear equations known as the ´load flow´. Given the operating conditions of the system, the load flow determines the voltages at all the nodes of the network, subsequently, any dependent variable may be calculated.

Each bus is characterized by four quantities [27]. The net real and reactive power injections, the voltage magnitude and the phase angle. Three types of buses are represented in the conventional load flow, and at each bus type, two of these four quantities are specified. It is necessary to select one bus, called the slack bus, at which both the voltage magnitude and the phase angle are specified. The need for this contraption arises because the real power losses, being a function of the solution voltages, are not known in advance. Since the power injections will be specified at all the other buses, it is necessary to have one bus (the slack bus) at which the real and reactive power generations are determined by the load flow. The remaining buses of the system are designated either as voltage controlled buses (also known as PV) or as load buses (also known as PQ). Real power injections and voltage magnitudes are specified at PV buses, whereas both real and reactive power injections are specified at PQ buses.

The real and reactive power injections into node i can be expressed as :

$$S_i = P_i + j\,Q_i = V_i\,I_i^* \tag{3.3}$$

or equivalently

$$P_i = \text{Real}\,(V_i\,I_i^*) \tag{3.4}$$

$$Q_i = \text{Imag}\,(V_i\,I_i^*) \tag{3.5}$$

The current entering node i is:

$$I_i = \sum_{k=1}^{N} Y_{ik}\,V_k \tag{3.6}$$

where N is the number of buses in the system and Y_{ik} is the ik element of the admittance matrix .

In rectangular coordinates, the nodal voltage at bus i is :

$$V_i = e_i + j\, f_i \tag{3.7}$$

By substituting into (3.4) and (3.5), we obtain:

$$P_i = \sum_{k=1}^{N} e_i \,(G_{ik} e_k - B_{ik} f_k) + f_i \,(G_{ik} f_k + B_{ik} e_k) \tag{3.8}$$

$$Q_i = \sum_{k=1}^{N} f_i \,(G_{ik} e_k - B_{ik} f_k) - e_i \,(G_{ik} f_k + B_{ik} e_k) \tag{3.9}$$

where $Y_{ik} = G_{ik} + j\, B_{ik}$

At load bus i :

specify P_i as given by (3.8)

specify Q_i as given by (3.9)

At voltage controlled bus i :

specify P_i as given by (3.8)

specify $|V_i|^2 = e_i^2 + f_i^2$

At the slack bus s :

specify $|V_s|^2 = e_s^2 + f_s^2$

Usually the slack bus is made the reference bus by setting the phase angle to zero, or equivalently setting $f_s = 0$.

All in all there are (2N-1) nonlinear equations in (2N-1) unknowns. Newton's method [27] transforms the non-linear load flow equations into a linear set of equations that must be solved iteratively until the solution converges to the desired accuracy.

This section has reviewed the load flow problem in sufficient detail for our purposes. However one cannot pass by without remarking that the load flow problem has been extensively tackled in the literature, with various methods aiming at reducing memory requirements and/or decreasing execution time.

3.3 REFORMULATING THE LOAD FLOW PROBLEM

Of particular relevance to the current discussion are papers [20] , [21] and the work of Jarjis [22] which layed the fundamentals of the theory about to be presented. The most recent contribution is the work done by Banakar

[23] where the approximation formulae were derived.

This section will be devoted to restating the load flow problem and establishing the fundamental tools necessary for deriving the approximation formulae. The formulae will be presented in section 3.4, their derivation has been postponed to the final section of this chapter so that the arduous mathematical manipulations will not obscure their elegance and simplicity.

Let N be the number of buses in the network

$\underline{e}$ $\quad (e_1, e_2, \ldots, e_N)^T$

$\underline{f}$ $\quad (f_1, f_2, \ldots, f_N)^T$

$\underline{x}$ $\quad (\underline{e}^T, \underline{f}^T)$

z_k $\quad$ net injection at bus k :

if bus k is PQ , the injections are P_k , Q_k

if bus k is PV , the injections are P_k , $|V_k|^2$

for the slack bus s , the injections are e_s^2 , f_s^2

$\underline{z} = F(\underline{x})$ load flow equations

Each component z_k of the vector $\underline{z}$ (i.e $P_k, Q_k, |V_k|^2$) can be expressed in a quadratic form

$$z_k = \underline{x}^T \underline{J}_{z_k} \underline{x} \qquad (3.10)$$

Where $\underline{J}_{z_k}$ is a (2N) * (2N) real symmetric matrix uniquely defined by the type of bus injection and the network structure. In case the slack bus imaginary voltage is set equal to zero, the dimension reduces to (2N-1) * (2N-1). Hence we may write :

$$\underline{z} = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_{2N-1} \end{bmatrix} = \begin{bmatrix} \underline{x}^T \underline{J}_{z_1} \underline{x} \\ \underline{x}^T \underline{J}_{z_2} \underline{x} \\ \vdots \\ \underline{x}^T \underline{J}_{z_{2N-1}} \underline{x} \end{bmatrix} \qquad (3.11)$$

Defining the matrix $\underline{L}(\underline{x})$ as :

$$\underline{L}(\underline{x}) = \begin{bmatrix} \underline{x}^T \underline{J}_{z_1} \\ \underline{x}^T \underline{J}_{z_2} \\ \vdots \\ \underline{x}^T \underline{J}_{z_{2N-1}} \end{bmatrix} \tag{3.12}$$

the load flow equations take the form ,

$$\underline{z} = \underline{L}(\underline{x}) \; \underline{x} \tag{3.13}$$

An interesting observation is that $\underline{L}(\underline{x})$ is equal to one-half the Jacobian matrix of the load flow equations. This can be easily verified by partial differentiation of $\underline{z}$ with respect to $\underline{x}$ in (3.11).

Dependent variables, $y(\underline{x})$, in a load flow problem include real losses , line flows , reactive generations , and load bus voltage magnitudes squared. All of these dependent load flow variables may be expressed in quadratic form :

$$y\,(\underline{x}) = \underline{x}^{T}\;\underline{R}\;\underline{x} \qquad (3.14)$$

Where $\underline{R}$ is a highly sparse constant matrix corresponding to the specified dependent variable y. Appendix A demonstrates a typical derivation of the $\underline{R}$ matrix.

3.4 THE APPROXIMATION FORMULAE

Expanding the dependent variable $y(\underline{x})$ in a Taylor series in $\underline{z}$ around some nominal (base case) voltage , $\underline{x}^{o}$, we get

$$y = y(\underline{x}^{o}) + \left(\frac{\partial y}{\partial \underline{z}}\right)\Bigg|_{\underline{x}=\underline{x}^{o}} \Delta\underline{z} + \frac{1}{2}\,\Delta\underline{z}^{T}\left(\frac{\partial^{2} y}{\partial \underline{z}^{2}}\right)\Bigg|_{\underline{x}=\underline{x}^{o}} \Delta\underline{z} + \ldots \qquad (3.15)$$

where $$\Delta\underline{z} = \underline{z} - \underline{z}^{o} \qquad (3.16)$$

and $$\underline{z}^{o} = \underline{L}\,(\underline{x}^{o})\;\underline{x}^{o} \qquad (3.17)$$

Then a linear approximation would comprise of the first two terms, whereas the quadratic approximation would include the first three terms.

The linear approximation formula for the dependent load flow variable $y(\underline{x})$ in terms of the specified injections $\underline{z}$ is given by

$$y = \underline{B}^T \underline{z} \tag{3.18}$$

where

$$\underline{B}^T = (\underline{x}^o)^T \underline{R} \; L^{-1} (\underline{x}^o) \tag{3.19}$$

The quadratic formula is given by

$$y = \underline{B}^T \underline{z} + \underline{z}^T \underline{C} \; \underline{z} \tag{3.20}$$

where $\underline{B}$ is given by (3.19), and

$$\underline{C} = \frac{1}{4} \left[\underline{L}^T (\underline{x}^o)\right]^{-1} \left[\underline{R} - \underline{J}\,(\underline{B})\right] \left[\underline{L}\,(\underline{x}^o)\right]^{-1} \tag{3.21}$$

3.5 DERIVATION OF THE APPROXIMATION FORMULAE

3.5.1 USEFUL PROPERTIES

Prior to proceeding with the formulae derivation two useful properties will be presented.

1. for any constant (2N-1) vector $\underline{B}^T = (B_1, B_2, \ldots, B_{2N-1})$

$$\underline{B}^T \underline{z} = \sum_{i=1}^{2N-1} B_i z_i \tag{3.22}$$

$$= \sum_{i=1}^{2N-1} B_i \underline{x}^T \underline{J}_{z_i} \underline{x} \tag{3.23}$$

$$= \underline{x}^T \left[\sum_{i=1}^{2N-1} B_i \underline{J}_{z_i} \right] \underline{x} \tag{3.24}$$

$$= \underline{x}^T \underline{J}(\underline{B}) \underline{x} \tag{3.25}$$

where $$\underline{J}(\underline{B}) = \sum_{i=1}^{2N-1} B_i \underline{J}_{z_i} \tag{3.26}$$

In (3.13) it was found that,

$$\underline{z} = \underline{L}(\underline{x}) \underline{x} \tag{3.27}$$

which when premultiplied by the $\underline{B}^T$ vector, gives

$$\underline{B}^T \underline{z} = \underline{B}^T \underline{L}(\underline{x}) \underline{x} \tag{3.28}$$

Comparing (3.25) to (3.28) we get;

$$\underline{x}^T \underline{J}(\underline{B}) = \underline{B}^T \underline{L}(\underline{x}) \tag{3.29}$$

Of particular interest is that $\underline{J}(\underline{B})$ is symmetric and is as sparse as the Jacobian matrix.

2. The partial derivative may be found by applying the chain rule as follows :

$$\frac{\partial \, y}{\partial \, \underline{z}} = \frac{\partial \, y}{\partial \, \underline{x}} \; \frac{\partial \, \underline{x}}{\partial \, \underline{z}} \qquad (3.30)$$

In (3.10) it was found that

$$z_i = \underline{x}^T \; \underline{J}_{z_i} \; \underline{x} \qquad (3.31)$$

Taking the partial derivative of (3.31) with respect to $\underline{x}$ we obtain,

$$\frac{\partial \, z_i}{\partial \, \underline{x}} = 2 \; \underline{x}^T \; \underline{J}_{z_i} \qquad (3.32)$$

which when carried out for i=1,...,N gives :

$$\frac{\partial \, \underline{z}}{\partial \, \underline{x}} = 2 \; \underline{L}(\underline{x}) \qquad (3.33)$$

where $\underline{L}(\underline{x})$ has been defined in (3.12) . The above equation

informs us that $\underline{L}(\underline{x})$ is equal to 1/2 the Jacobian .
Inverting (3.33)

$$\frac{\partial \underline{x}}{\partial \underline{z}} = \frac{1}{2} \underline{L}(\underline{x})^{-1} \qquad (3.34)$$

and substituting in (3.30)

$$\Longrightarrow \quad \frac{\partial y}{\partial \underline{z}} = \frac{1}{2} \frac{\partial y}{\partial \underline{x}} \underline{L}(\underline{x})^{-1} \qquad (3.35)$$

3.5.2 FORMULAE DERIVATION

Having presented the above properties we are now in a good position to proceed with the Taylor series expansion.

$$y = y(\underline{x}^{o}) + \left.\frac{\partial y}{\partial \underline{z}}\right|_{\underline{x}=\underline{x}^{o}} \Delta\underline{z} + \frac{1}{2} \Delta\underline{z}^{T} \left.\frac{\partial^{2} y}{\partial \underline{z}^{2}}\right|_{\underline{x}=\underline{x}^{o}} \Delta\underline{z} + \ldots \qquad (3.36)$$

Each term of the series will be examined separately.

(1) <u>first term</u>

$$T_1 = y(\underline{x}^o) \tag{3.37}$$

Therefore T_1 is simply (3.14) evaluated at base case voltages.

$$= \underline{x}^T \underline{R} \underline{x} \Big|_{\underline{x}=\underline{x}^o} \tag{3.38}$$

$$= (\underline{x}^o)^T \underline{R} \underline{x}^o \tag{3.39}$$

(2) <u>second term</u>

$$T_2 = \frac{\partial y}{\partial \underline{z}} \Bigg|_{\underline{x}=\underline{x}^o} \underline{\Delta z} \tag{3.40}$$

In the previous subsection, property (2) showed that the partial derivative of (3.40) is given by;

$$\frac{\partial y}{\partial \underline{z}} = \frac{1}{2} \frac{\partial y}{\partial \underline{x}} \underline{L}^{-1}(\underline{x}) \tag{3.41}$$

Moreover, the partial derivative in (3.41) may be easily determined by substituting for y as given by (3.14).

$$\Longrightarrow \quad \frac{\partial y}{\partial \underline{x}} = \frac{\partial}{\partial \underline{x}} \left(\underline{x}^T \underline{R} \, \underline{x} \right) \tag{3.42}$$

$$= 2 \, \underline{x}^T \underline{R} \tag{3.43}$$

Substituting (3.41) and (3.43) into (3.40), we get

$$T_2 = \underline{x}^T \underline{R} \, \underline{L}(\underline{x})^{-1} \Big|_{\underline{x}=\underline{x}^o} \underline{\Delta z} \tag{3.44}$$

$$T_2 = \underline{B}(\underline{x}^o)^T \underline{\Delta z} \tag{3.45}$$

where

$$\underline{B}(\underline{x}^o)^T = (\underline{x}^o)^T \underline{R} \, \underline{L}(\underline{x}^o)^{-1} \tag{3.46}$$

(3) <u>third term</u>

$$T_3 = \frac{1}{2} \underline{\Delta z}^T \frac{\partial^2 y}{\partial \underline{z}^2} \Bigg|_{\underline{x}=\underline{x}^o} \underline{\Delta z} \tag{3.47}$$

Start by examining $\dfrac{\partial^2 y}{\partial \underline{z}^2} \, \underline{\Delta z}$:

$$\frac{\partial^2 y}{\partial \underline{z}^2} \, \underline{\Delta z} = \frac{\partial T_2}{\partial \underline{z}} \qquad (3.48)$$

$$= \frac{\partial T_2}{\partial \underline{x}} \quad \frac{\partial \underline{x}}{\partial \underline{z}} \qquad (3.49)$$

$$= \frac{\partial T_2}{\partial \underline{x}} \left[\frac{1}{2} \, \underline{L}(\underline{x})^{-1} \right] \qquad (3.50)$$

(3.50) is obtained by replacing the partial derivative of $\underline{x}$ in (3.49) with expression (3.34).

Moreover T_2 , in (3.50), has already been found in (3.44) to be,

$$T_2 = \underline{B}(\underline{x})^T \quad \underline{\Delta z} \qquad (3.51)$$

with

$$\underline{B}(\underline{x})^T = \underline{x}^T \quad \underline{R} \quad \underline{L}(\underline{x})^{-1} \qquad (3.52)$$

In order to compute the partial derivative of T_2 in (3.50),

$$\frac{\partial T_2}{\partial \underline{x}} = \frac{\partial \underline{B}(\underline{x})^T}{\partial \underline{x}} \Delta \underline{z} \tag{3.53}$$

consider a differential change in T_2, or equivalently in $\underline{B}(\underline{x})^T$;

$$d\underline{B}(\underline{x})^T = d(\underline{x}^T) \, \underline{R} \, \underline{L}(\underline{x})^{-1} + \underline{x}^T \, \underline{R} \, d\left[\underline{L}(\underline{x})^{-1}\right] \tag{3.54}$$

Noting that,

$$d(\underline{L}(\underline{x})^{-1}) = - \underline{L}(\underline{x})^{-1} \, d\underline{L}(\underline{x}) \, \underline{L}(\underline{x})^{-1} \tag{3.55}$$

(3.54) may be rewritten as ;

$$d\underline{B}(\underline{x})^T = d(\underline{x}^T) \, \underline{R} \, \underline{L}(\underline{x})^{-1} - \underline{x}^T \, \underline{R} \, \underline{L}(\underline{x})^{-1} \, d(\underline{L}(\underline{x})) \, \underline{L}(\underline{x})^{-1} \tag{3.56}$$

Also, since we know that

$$\underline{B}(\underline{x})^T = \underline{x}^T \, \underline{R} \, \underline{L}(\underline{x})^{-1} \tag{3.57}$$

(3.56) may be rewritten as;

$$d\underline{B}(\underline{x})^T = d(\underline{x}^T) \, \underline{R} \, \underline{L}(\underline{x})^{-1} - \underline{B}(\underline{x})^T \, d\left[\underline{L}(\underline{x})\right] \, \underline{L}(\underline{x})^{-1} \tag{3.58}$$

Recalling property (1) where,

$$\underline{B}^T(\underline{x}) \; \underline{L}(\underline{x}) = \underline{x}^T \; \underline{J}(\underline{B}) \tag{3.59}$$

(3.58) can be simplified to :

$$d\underline{B}^T(\underline{x}) = d(\underline{x}^T) \; \underline{R} \; \underline{L}^{-1}(\underline{x}) - d(\underline{x}^T) \; \underline{J}(\underline{B}) \; \underline{L}^{-1}(\underline{x}) \tag{3.60}$$

$$\Longrightarrow d\underline{B}^T(\underline{x}) = d(\underline{x})^T \left[\underline{R} - \underline{J}(\underline{B}) \right] \underline{L}^{-1}(\underline{x}) \tag{3.61}$$

Returning to the partial derivative in (3.53), it may be rewritten using (3.61) as,

$$\frac{\partial T_2}{\partial \underline{x}} = \left[\underline{L}^{-1}(\underline{x}) \right]^T \left[\underline{R} - \underline{J}(\underline{B}) \right] \Delta\underline{z} \tag{3.62}$$

and substituting (3.62) into (3.50) we obtain:

$$\frac{\partial T_2}{\partial \underline{z}} = \frac{1}{2} \left[\underline{L}^{-1}(\underline{x}) \right]^T \left[\underline{R} - \underline{J}(\underline{B}) \right] \underline{L}^{-1}(\underline{x}) \; \Delta\underline{z} \tag{3.63}$$

Finally, by replacing (3.63) into (3.47), we get;

$$T_3 = \frac{1}{4} \, \Delta \underline{z}^T \left[\underline{L}(\underline{x})^{-1} \right]^T \left[\underline{R} - \underline{J}(\underline{B}) \right] \underline{L}(\underline{x})^{-1} \, \Delta \underline{z} \Big|_{\underline{x}=\underline{x}^o}$$

(3.64)

$$\Longrightarrow T_3 = \frac{1}{4} \, \Delta \underline{z}^T \left[\underline{L}(\underline{x}^o) \right]^{-T} \left[\underline{R} - \underline{J}(\underline{B}) \right] \left[\underline{L}(\underline{x}^o) \right]^{-1} \Delta \underline{z}$$

(3.65)

Define the $\underline{C}$ matrix to be :

$$\underline{C} = \frac{1}{4} \left[\underline{L}(\underline{x}^o) \right]^{-T} \left[\underline{R} - \underline{J}(\underline{B}) \right] \left[\underline{L}(\underline{x}^o) \right]^{-1} \quad (3.66)$$

Having evaluated each of the first three terms of the Taylor series expansion, equation (3.36) may be reexpressed as :

$$y = y(\underline{x}^o) + \underline{B}^T(\underline{x}^o) \, (\underline{z} - \underline{z}^o) + (\underline{z} - \underline{z}^o) \, \underline{C} \, (\underline{z} - \underline{z}^o)$$

(3.67)

Since $\underline{B}^T(\underline{x}^o) \, \underline{z}^o = \underline{y}(\underline{x}^o)$, (3.67) becomes :

$$y = \underline{B}^T(\underline{x}^o) \, \underline{z} + (\underline{z} - \underline{z}^o) \, \underline{C} \, (\underline{z} - \underline{z}^o) \quad (3.68)$$

The first term of (3.68) gives the linear approximation. Substituting the value of $\underline{C}$ (3.66) into (3.68), and noticing that:

$$\underline{J}(\underline{B})\ \underline{x}^{o} = \underline{L}^{T}(\underline{x}^{o})\ \underline{B}(\underline{x}^{o})$$

$$= \underline{L}^{T}(\underline{x}^{o})\ \underline{L}^{-T}(\underline{x}^{o})\ \underline{R}\ \underline{x}^{o}$$

$$= \underline{R}\ \underline{x}^{o} \qquad (3.69)$$

we find that ;

$$\underline{z}\ \underline{C}\ \underline{z}^{o} = \underline{z}^{o}\ \underline{C}\ \underline{z} = \underline{z}^{o}\ \underline{C}\ \underline{z}^{o} = 0$$

Thus, (3.68) reduces to :

$$y = \underline{B}^{T}(\underline{x}^{o})\ \underline{z} + \underline{z}\ \underline{C}\ \underline{z} \qquad (3.70)$$

which is the quadratic form.

CHAPTER 4

THE PROPOSED EQUIVALENCING METHOD

4.1 MOTIVATION

Consider a power system , fig 6 , that has been subdivided into internal and external areas.

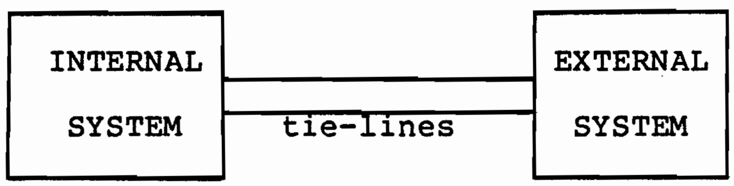


FIG 6

Given a set of base case injections , say , $\underline{Z} = (\underline{Z}internal , \underline{Z}external)$ where $\underline{Z}internal$ denotes the internal injections and $\underline{Z}external$ denotes the external injections , a load flow for the complete system will yield a solution voltage , say , $\underline{V} = (\underline{V}internal , \underline{V}external)$ where once again the internal and external bus voltages have been grouped separately .

Isolate the internal system by severing the tie-lines at the boundary buses . Using injections $\underline{Z}internal$

a load flow for the isolated internal system will render a solution voltage, say, $\underline{V}$isolated that generally differs drastically from $\underline{V}$internal . Therefore, we reach the obvious conclusion that isolating the study system, is not a valid equivalencing method . The dependence of the internal system on the external system is exemplified by the load flow problem where all (2N-1) equations must be solved simultaneously .

To amend the discrepancy, one can augment the boundary bus injections of the isolated internal system by the corresponding tie-line flows, as given by the complete system base case load flow. The isolated internal system load flow will have a solution voltage, say, $\underline{V}$isolated that should be identical to $\underline{V}$internal.

We should then apply the following recipe for obtaining an exact equivalent ;

1. Run a load flow for the complete system and determine the tie-line flows.
2. At each internal system boundary bus remove the tie-lines and augment the injections by the net tie-line flows.

So far the feasibility of simulating the effect of the eliminated external system by supplementing the boundary injections has been demonstrated. However, the approach pursued above is not an appealing method for equivalencing . Since after all,it requires solving a complete system load flow whenever a change in operating conditions occurs. A more captivating approach is one where the effect of the external system is reduced to additional boundary bus injections in a fashion that lends itself to a simpler updating as a function of the operating conditions. This is where the approximation formulae, introduced in the preceeding chapter, step into the picture.

The intense mathematical nature of Chapter 3 was unavoidable for attaining a clear understanding of the approximation formulae. That done, it is hoped that the quantities already defined will acquire a more practical meaning as we go along.

It has been mentioned in Chapter 3 that the approximation formulae, which are based on a Taylor series expansion, relate an arbitrary load flow variable, y, to the independent injections, $\underline{z}$, of a general load flow problem . Translated into our objective it conveys that the real and reactive tie-line flows into the boundary buses may be expanded in a Taylor series about a known base case. Subsequently, the tie-line flows for any other injection vector may be easily approximated.

As is to be expected, the more terms included in the Taylor series expansion the more accurate is the expression. We shall confine our attention in this work to the linear and quadratic expressions, it is anticipated that the computational effort for finding higher order terms may not compete with the time required for solving the complete system load flow. Actually most of the weight of the Taylor series is embeded in the first term, with latter terms carrying progressively less weights.

4.2 APPROXIMATING THE TIE-LINE FLOWS

The first step in the process of constructing the equivalent is to find the $\underline{B}$ vectors for the linear approximation, and the $\underline{C}$ matrices if the quadratic approximation is desired, corresponding to the real and reactive power flows at the boundary buses in each tie-line of fig 6 . The $\underline{B}$ vector and the $\underline{C}$ matrix were derived in (3.46) and (3.66) respectively,and are repeated in (4.1) and (4.2).

$$\underline{B}^T = (\underline{x}^o)^T \; \underline{R} \; \left[\underline{L}(\underline{x}^o)\right]^{-1} \tag{4.1}$$

$$\underline{C} = \frac{1}{4} \left[\underline{L}(\underline{x}^{o})^{T} \right]^{-1} \left[\underline{R} - \underline{J}(\underline{B}) \right] \left[\underline{L}(\underline{x}^{o}) \right]^{-1} \quad (4.2)$$

Since we are working with a Taylor series expansion, knowledge of a base case for the complete network is a prerequisite.

The base case voltages are denoted by $\underline{x}^{o}$.

It may be recalled that ;

$\underline{B}$ is a (2N-1) vector.

$\underline{C}$ is a matrix of order (2N-1).

$\underline{L}(\underline{x})$ is 1/2 the Jacobian .

$\underline{R}$ is a matrix that depends on the network structure and the dependent variable .

$\underline{J}(\underline{B})$ defined in equation (3.26) , it depends on $\underline{B}$ and $\underline{J}_{z_i}$.

For a specified injection $\underline{z}$, it is relatively straightforward to calculate each of the real and reactive tie-line power flow approximations using formulae (3.18) and (3.20), repeated below ;

The linear expression

$$y = \underline{B}^T \underline{z} \qquad (4.3)$$

The quadratic expression

$$y = \underline{B}^T \underline{z} + \underline{z}^T \underline{C} \underline{z} \qquad (4.4)$$

This completed, the next step is to adjoin the above injections to the boundary buses .

In the case where there is more than one tie-line emanating from a boundary bus, it is possible to proceed as outlined above; i.e. to find the $\underline{R}$ matrices corresponding to each of the real and reactive tie-line power flows and subsequently determine all the $\underline{B}$ vectors (and $\underline{C}$ matrices). A more computationally efficient alternative is to find the single equivalent $\underline{R}$ matrix corresponding to the real power flow and the single equivalent $\underline{R}$ matrix corresponding to the reactive power flow for these tie-lines, and use it to determine the equivalent $\underline{B}$ vectors (and $\underline{C}$ matrices). To reexpress this more tangibly, suppose that at a particular boundary bus there are L tie-lines. The former approach will result in 2L $\underline{B}$ vectors (one $\underline{B}$ vector for the real power and one $\underline{B}$ vector for the reactive power corresponding to each and every tie-line). Whereas, the latter approach will yield 2 $\underline{B}$ vectors only!

4.3 AUGMENTING THE BOUNDARY INJECTIONS

The procedure will first be demonstrated on a particular example, then general conclusions based on the insight gained will be stated.

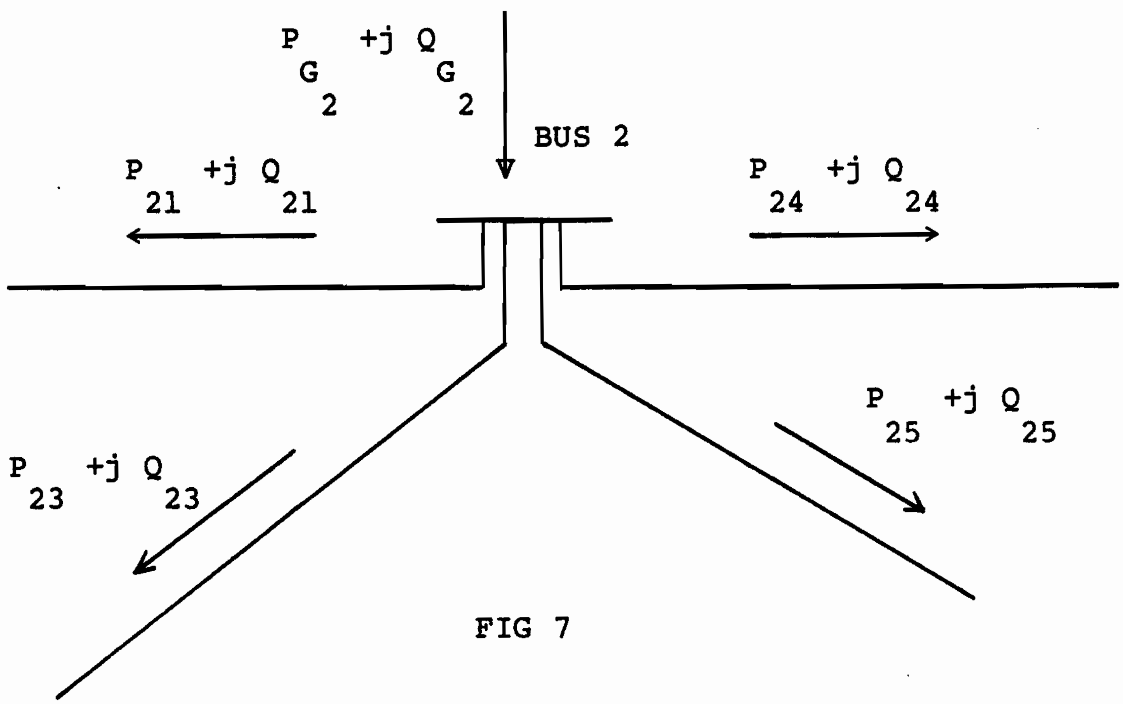


FIG 7

Consider a 5 bus system where the internal system consists of buses 1, 2 and 3, whereas buses 4 and 5 comprise the external system. Fig 7 singles out bus 2 of this system.

The objective is to consider in detail how one should handle the tie-line flows, when external buses 4 and 5 are removed, so that the internal system conditions remain unchanged.

<u>case a :</u> Bus 2 is a PV bus

Without loss of generality it will be assumed that there is no load at bus 2. Let the subscript R denote load flow quantities obtained after the elimination of external buses 4 and 5.

If the reduced model were exact all load flow variables should remain unchanged ;

$$\Longrightarrow \quad P_{21_R} = P_{21} \qquad Q_{21_R} = Q_{21} \tag{4.5}$$

$$P_{23_R} = P_{23} \qquad Q_{23_R} = Q_{23}$$

Since bus 2 is a PV bus only the real power injection is specified . The flows P_{24} and P_{25} have already been evaluated using the approximation formulae, thus we may write

$$P_{G_{2R}} = P_{G_2} - P_{24} - P_{25} \tag{4.6}$$

This injection will ensure that all real and reactive flows in the reduced network would be identical to the corresponding power flows in the complete system.

Let us now examine the reactive generation at bus 2 (which is a dependent variable of the load flow problem). Since the sum of reactive powers flowing into a bus must add up to zero , we may write

$$Q_{G_{2R}} = Q_{21_R} + Q_{23_R} \tag{4.7}$$

But from equation (4.5),this implies that;

$$Q_{G_{2R}} = Q_{21} + Q_{23} \tag{4.8}$$

however in the complete system

$$Q_{G_2} = Q_{21} + Q_{23} + Q_{24} + Q_{25} \tag{4.9}$$

Therefore , by comparing (4.8) and (4.9) we discern that ;

$$Q_{G_2} = Q_{G_{2R}} + Q_{24} + Q_{25} \tag{4.10}$$

which states that the reactive generation at bus 2 in the reduced load flow will differ from the exact value by $(Q_{24} + Q_{25})$.

To summarize the above observations for a PV bus :

i) The tie-line real power flows must be included as an additional boundary bus injection , this will guarantee that the real and reactive power flows in the equivalent model will be exact. However, this will not result in the exact reactive generation at that bus .

ii) To obtain the exact reactive generation one must add the reactive tie-line flow approximations to the reactive generation obtained from the reduced load flow . The point being emphasized is that at a PV bus the reactive tie-line flow approximations must not be treated as an additional load at that bus .

<u>case b</u> Bus 2 is a PQ bus

In this case both the real and reactive tie-line flow approximations are added respectively to the real and reactive loads at that bus .

4.4 ILLUSTRATION

Having described in general terms how to obtain the equivalent , this section will implement the method on the 5 bus system shown in fig 8 .

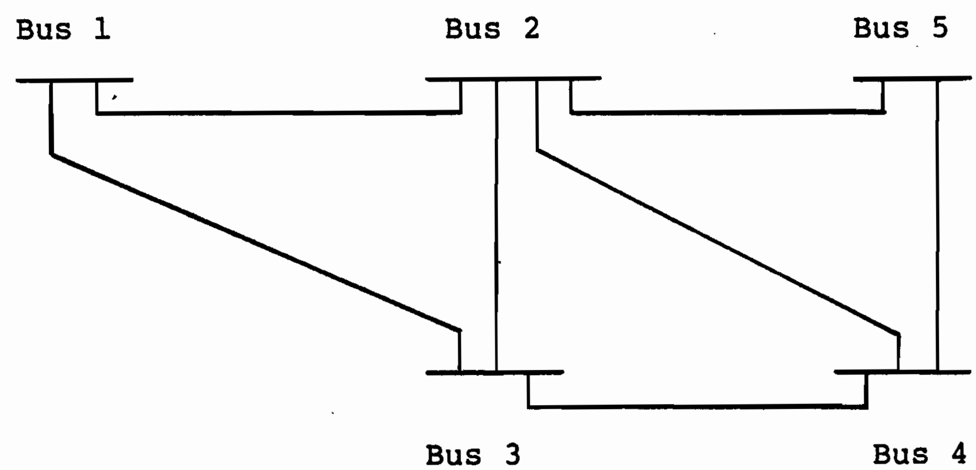


FIG 8 5 Bus System

The line data for the 5 bus system is provided in table 1 .

Table 1

line	impedance	1/2 charging admittance
1-2	.020 +j .060	.030
1-3	.080 +j .240	.025
2-3	.060 +j .180	.020
2-4	.060 +j .180	.020
2-5	.040 +j .120	.015
3-4	.010 +j .030	.010
4-5	.080 +j .240	.025

The bus data is provided in table 2.

Table 2

BUS	TYPE	VOLTAGE MAGNITUDE	VOLTAGE ANGLE	GENERATION REAL	GENERATION REACTIVE	LOAD REAL	LOAD REACTIVE
1	slack	1.06	0.	?	?	0.	0.
2	PV	1.05	?	.692	?	.200	.100
3	PV	1.04	?	.527	?	.450	.150
4	PQ	?	?	0.	0.	.400	.050
5	PQ	?	?	0.	0.	.600	.100

The internal system comprises of buses 1,2 and 3 with buses 2 and 3 being boundary buses . Accordingly the three tie-lines are 2-4 , 2-5 and 3-4 . The base case load flow , as provided by the load flow program included in Appendix C , is quoted in table 3.

Table 3

		VOLTAGE MAGNITUDE	VOLTAGE ANGLE	GENERATION REAL	GENERATION REACTIVE	DEMAND REAL	DEMAND REACTIVE
BUS	1	1.060	0.0	0.448	0.058	0.0	0.0
	TO BUS 3			0.158	0.010		
	TO BUS 2			0.289	0.048		
BUS	2	1.050	-0.81	0.692	0.043	0.200	0.100
	TO BUS 5			0.494	0.055		
	TO BUS 4			0.172	-0.001		
	TO BUS 3			0.114	-0.001		
	TO BUS 1			-0.288	-0.110		
BUS	3	1.040	-1.82	0.527	0.033	0.450	0.150
	TO BUS 4			0.347	-0.017		
	TO BUS 2			-0.113	-0.041		
	TO BUS 1			-0.157	-0.059		
BUS	4	1.037	-2.38	0.0	0.0	0.400	0.050
	TO BUS 5			0.117	-0.010		
	TO BUS 3			-0.346	-0.002		
	TO BUS 2			-0.171	-0.038		
BUS	5	1.024	-3.81	0.0	0.0	0.600	0.100
	TO BUS 4			-0.115	-0.040		
	TO BUS 2			-0.485	-0.060		

TOTAL SYSTEM LOSS = 0.017

step 1 Find the $\underline{B}$ vector and $\underline{C}$ matrix

The following $\underline{B}$ vectors need to be found :

$\underline{B}_{2-4}$, $\underline{B}_{2-5}$, $\underline{B}_{3-4}$ corresponding to real power

$\underline{B}_{2-4}$, $\underline{B}_{2-5}$, $\underline{B}_{3-4}$ corresponding to reactive power

If the quadratic approximation is also desired, then in addition to finding the $\underline{B}$ vectors, the corresponding $\underline{C}$ matrices must also be determined . The approximation program used in this work has been inserted in Appendix B along with a detailed simulation of this study case. Accordingly all the $\underline{B}$ vectors and $\underline{C}$ matrices are explicitly included in Appendix B.

Having found the $\underline{B}$ vectors (and the $\underline{C}$ matrices) equations (4.3) and (4.4) may be applied to find the approximate power flows corresponding to a specified injection vector. If the injection vector is identical to the base case injection vector, one would expect that the power flows must also be identical to the base case flows. To convince oneself that the Taylor expansion satisfies this, it is sufficient to set $\Delta\underline{Z} = 0$ in equation (3.15). If the injections were different from the base case injections, the power flows in the reduced model would be an approximation to the actual power flows. Typical of a Taylor series expansion, the further removed the injections are from the base case injections the larger is the error inherent in the approximation.

To continue with the demonstration let the injection vector be the base case injection. Both linear and quadratic approximation formulae give (Appendix B) ;

$$P_{24} = .172 \qquad Q_{24} = -\ .001$$

$$P_{25} = .494 \qquad Q_{25} = \ .055$$

$$P_{34} = .347 \qquad Q_{34} = -\ .017$$

step 2 Augment the boundary injections

Proceed as described in section 4.3 . For the current 5 bus example, both boundary buses are of the PV type. Thus only the real power injection needs to be updated prior to running the reduced system load flow. The reduced system line data is summarized in table 4.

Table 4

line	impedance	1/2 charging admittance
1-2	.020 +j .060	.030
1-3	.080 +j .240	.025
2-3	.060 +j .180	.020

The reduced system bus data is provided in table 5 .

Table 5

		V O L T A G E		GENERATION		LOAD	
BUS	TYPE	MAGNITUDE	ANGLE	REAL	REACTIVE	REAL	REACTIVE
1	slack	1.06	0.	?	?	0.	0.
2	PV	1.05	?	.692	?	.866	.100
3	PV	1.04	?	.527	?	.797	.150

The resulting reduced system load flow is summarized in table 6 .

Table 6

		VOLTAGE MAGNITUDE	VOLTAGE ANGLE	GENERATION REAL	GENERATION REACTIVE	DEMAND REAL	DEMAND REACTIVE
BUS	1	1.060	0.0	0.448	0.058	0.0	0.0
	TO BUS 3			0.158	0.010		
	TO BUS 2			0.289	0.048		
BUS	2	1.050	-0.81	0.692	-0.011	0.866	0.100
	TO BUS 3			0.114	-0.001		
	TO BUS 1			-0.288	-0.110		
BUS	3	1.040	-1.82	0.527	0.050	0.797	0.150
	TO BUS 2			-0.113	-0.041		
	TO BUS 1			-0.157	-0.059		

When table 6 is compared to table 3 it is noted that all load flow variables are identical (just as was expected), except for Q_{G2} and Q_{G3}.

This discrepancy at boundary PV buses was pointed out in section 4.3; it may be rectified by augmenting the generations by the approximate tie-line flows (Appendix B).

$$Q_{G_{2R}} + Q_{24} + Q_{25} = -0.011 - 0.001 + 0.055 = 0.043$$

$$Q_{G_{3R}} + Q_{34} = 0.050 - 0.017 = 0.033$$

which are identical to the corresponding values in table 3.

4.5 UPDATING THE EQUIVALENT

The ease with which the new tie-line flows may be found following a change in injections is evident from (4.3) and (4.4). One is equally interested, as in security studies, in simulating internal network changes (i.e. outages). It may be recalled that the $\underline{B}$ vector and $\underline{C}$ matrix depend on the network configuration and the type of injection being approximated (and, of course, on the base case). Accordingly both the $\underline{B}$ vector and $\underline{C}$ matrix will change if the network structure changes. In principle,to find the new equivalent, one would have to re-start the involved and time consuming computation of the $\underline{B}$ vectors and the $\underline{C}$ matrices.

We have investigated whether there exists a simple prescription for updating the approximations. In this section two methods for updating the linear approximations will be discussed.

4.5.1 METHOD 1

Consider the power network in fig 9 which has been subdivided into an internal system (whose quantities are denoted by subscript 2), and an external system (whose quantities are denoted by subscript 1).

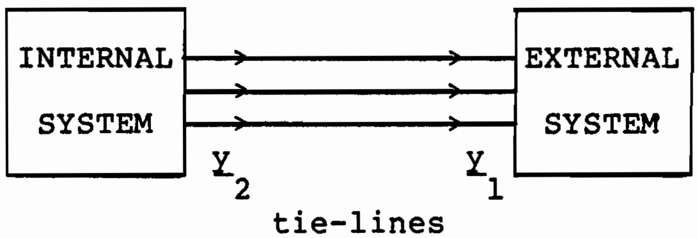


FIG 9

$\underline{y}_1$ and $\underline{y}_2$ are the vectors of power flows at the tie-line extremities, they may be written as ;

$$\underline{y}_2 = \underline{G}_2(\underline{x}_1, \underline{x}_2) \tag{4.11}$$

$$\underline{y}_1 = \underline{G}_1(\underline{x}_1, \underline{x}_2) \tag{4.12}$$

Where $\underline{x}_1$ is the vector of internal system voltages and $\underline{x}_2$ is the vector of external system voltages. Since $\underline{y}_1$ and $\underline{y}_2$ are dependent load flow variables, they may be expressed in the form of (3.14). Consequently, $\underline{G}_1$ and $\underline{G}_2$ are functions of the network topology and the voltages.

The external system load flow equations are

$$\underline{z}_1 = \underline{F}_1(\underline{x}_1) + \underline{A}_1 \underline{y}_1 \qquad (4.13)$$

and the internal system load flow equations are

$$\underline{z}_2 = \underline{F}_2(\underline{x}_2) + \underline{A}_2 \underline{y}_2 \qquad (4.14)$$

Where $\underline{z}_1$ is the vector of external system injections. $\underline{F}_1(\underline{x}_1)$ is the right hand side of the load flow equations of of the isolated (no interconnections) external system , this is clearly independent of $\underline{x}_2$. The matrix $\underline{A}_1$ is constant and contains all zeroes except for a -1 in the jk elements,where j is the row number of a boundary bus power injection in $\underline{z}_1$, and k is the row number of the tie-line power flow at that bus in $\underline{y}_1$ (recall that at a PV bus no reactive power is specified, accordingly the corresponding entry would be zero). Therefore the vector $\underline{A}_1 \underline{y}_1$ is the set of injections that augment the boundary injections in $\underline{z}_1$.

Likewise , $\underline{z}_2$ is the vector of internal system injections . $\underline{F}_2(\underline{x}_2)$ is the right hand side of the isolated internal system load flow equations , and is independent of $\underline{x}_1$. The matrix $\underline{A}_2$ is constant and contains a +1 in the jk elements, where j is the row number of a boundary bus power injection in $\underline{z}_2$, and k is the row number of the tie-line power flow at that bus in $\underline{y}_2$ (the same remark concerning PV buses holds true).

When solved simultaneously,equations (4.11),(4.12), (4.13) and (4.14) yield the exact complete system load flow.

From (4.3) , we know that a dependent variable y may be expressed as :

$$y = \underline{B}^T \underline{z} = \underline{B}_1^T \underline{z}_1 + \underline{B}_2^T \underline{z}_2 \qquad (4.15)$$

The vector $\underline{B}$ can be found from the inverse Jacobian of the whole system as was shown in Chapter 3 . However , we would like to express it in terms of the Jacobian of the internal network so that it will be easier to modify in the case of internal contingencies.

Notation

$$\underline{J}_1 = \frac{\partial \underline{F}_1}{\partial \underline{x}_1} \qquad \underline{J}_2 = \frac{\partial \underline{F}_2}{\partial \underline{x}_2}$$

$$\underline{J}_3 = \frac{\partial \underline{G}_1}{\partial \underline{x}_1} \qquad \underline{J}_4 = \frac{\partial \underline{G}_1}{\partial \underline{x}_2}$$

$$\underline{J}_5 = \frac{\partial \underline{G}_2}{\partial \underline{x}_1} \qquad \underline{J}_6 = \frac{\partial \underline{G}_2}{\partial \underline{x}_2}$$

The incremental equations of (4.11), (4.12),(4.13), (4.14) are :

$$\Delta\underline{y}_2 = \underline{J}_5 \, \Delta\underline{x}_1 + \underline{J}_6 \, \Delta\underline{x}_2 \tag{4.16}$$

$$\Delta\underline{y}_1 = \underline{J}_3 \, \Delta\underline{x}_1 + \underline{J}_4 \, \Delta\underline{x}_2 \tag{4.17}$$

$$\Delta\underline{z}_1 = \underline{J}_1 \, \Delta\underline{x}_1 + \underline{A}_1 \, \Delta\underline{y}_1 \tag{4.18}$$

$$\Delta\underline{z}_2 = \underline{J}_2 \, \Delta\underline{x}_2 + \underline{A}_2 \, \Delta\underline{y}_2 \tag{4.19}$$

Before proceeding it is important to note that both $\underline{J}_1$ and $\underline{J}_2$ cannot be nonsingular. If the slack bus is chosen to be in the internal system (as is reasonable), then $\underline{J}_1$ will be singular and $\underline{J}_2$ will be nonsingular. One way to perceive this is to recall that $\underline{F}_1(\underline{x}_1)$ is the right hand side of the load flow equations for the external system without interconnections. The absence of a slack bus in the external system implies that the real power generation must be specified at every bus, which in turn dictates that the the system is over-specified (i.e. no bus accounts for the real losses in the system),or in other words the Jacobian J_1 is singular. By reexamining the formulae that determine the the elements of a Jacobian [26] one can readily verify that the elements of $(\underline{J}_1+\underline{A}_1 \underline{J}_3)$ are included among the elements of the complete system Jacobian. In fact, it is possible to rearrange the elements of the complete system Jacobian (by renumbering the buses) so that the matrix $(\underline{J}_1+\underline{A}_1 \underline{J}_3)$ appears in the first principal block. Furthermore,since the complete system Jacobian is invertible, it is a necessary condition that $(\underline{J}_1+\underline{A}_1 \underline{J}_3)$ also be invertible.

The objective is to solve the incremental equations for an expression that relates the boundary flows to the injections and which is independent of the voltages. Substituting (4.17) into (4.18) yields ;

$$\underline{\Delta z}_1 = \underline{J}_1 \, \underline{\Delta x}_1 + \underline{A}_1 \, (\underline{J}_3 \, \underline{\Delta x}_1 + \underline{J}_4 \, \underline{\Delta x}_2) \tag{4.20}$$

$$\Longrightarrow \quad \underline{\Delta z}_1 = (\underline{J}_1 + \underline{A}_1 \underline{J}_3) \, \underline{\Delta x}_1 + \underline{A}_1 \underline{J}_4 \, \underline{\Delta x}_2 \tag{4.21}$$

$$\Longrightarrow \quad \underline{\Delta x}_1 = (\underline{J}_1 + \underline{A}_1 \underline{J}_3)^{-1} \, (\underline{\Delta z}_1 - \underline{A}_1 \underline{J}_4 \, \underline{\Delta x}_2) \tag{4.22}$$

Solving (4.19) for $\underline{\Delta x}_2$,

$$\Longrightarrow \quad \underline{\Delta x}_2 = \underline{J}_2^{-1} \, (\underline{\Delta z}_2 - \underline{A}_2 \, \underline{\Delta y}_2) \tag{4.23}$$

and substituting (4.22) and (4.23) into (4.16), we get ;

$$\underline{\Delta y}_2 = \underline{J}_5 \, (\underline{J}_1 + \underline{A}_1 \underline{J}_3)^{-1} \left[\underline{\Delta z}_1 - \underline{A}_1 \underline{J}_4 \underline{J}_2^{-1} \, (\underline{\Delta z}_2 - \underline{A}_2 \, \underline{\Delta y}_2) \right]$$

$$+ \underline{J}_6 \underline{J}_2^{-1} \, (\underline{\Delta z}_2 - \underline{A}_2 \, \underline{\Delta y}_2) \tag{4.24}$$

Transfering all terms containing $\underline{\Delta y}_2$ to the left hand side ;

$$\left[\underline{I} - \underline{J}_5 \, (\underline{J}_1 + \underline{A}_1 \underline{J}_3)^{-1} \underline{A}_1 \underline{J}_4 \underline{J}_2^{-1} \underline{A}_2 + \underline{J}_6 \underline{J}_2^{-1} \underline{A}_2 \right] \underline{\Delta y}_2$$

$$= \underline{J}_5 (\underline{J}_1 + \underline{A}_1 \underline{J}_3)^{-1} \Delta z_1 + \left[\underline{J}_6 \underline{J}_2^{-1} - \underline{J}_5 (\underline{J}_1 + \underline{A}_1 \underline{J}_3)^{-1} \underline{A}_1 \underline{J}_4 \underline{J}_2^{-1} \right] \Delta z_2 \tag{4.25}$$

Define ;

$$\underline{J}_7 = \underline{J}_5 (\underline{J}_1 + \underline{A}_1 \underline{J}_3)^{-1} \underline{A}_1 \underline{J}_4 \tag{4.26}$$

$$\underline{J}_8 = \underline{J}_5 (\underline{J}_1 + \underline{A}_1 \underline{J}_3)^{-1} \tag{4.27}$$

and substitute $\underline{J}_7$ and $\underline{J}_8$ in (4.25) to get

$$(\underline{I} - \underline{J}_7 \underline{J}_2^{-1} \underline{A}_2 + \underline{J}_6 \underline{J}_2^{-1} \underline{A}_2) \Delta \underline{y}_2 = \underline{J}_8 \Delta z_1 + (\underline{J}_6 \underline{J}_2^{-1} - \underline{J}_7 \underline{J}_2^{-1}) \Delta z_2 \tag{4.28}$$

$$\Longrightarrow \left[\underline{I} + (\underline{J}_6 - \underline{J}_7) \underline{J}_2^{-1} \underline{A}_2 \right] \Delta \underline{y}_2 = \underline{J}_8 \Delta z_1 + (\underline{J}_6 - \underline{J}_7) \underline{J}_2^{-1} \Delta z_2 \tag{4.29}$$

Define ;

$$\underline{J}_9 = \underline{J}_6 - \underline{J}_7 \tag{4.30}$$

$$\underline{J}_{10} = (\underline{I} + \underline{J}_9 \underline{J}_2^{-1} \underline{A}_2) \tag{4.31}$$

By substituting $\underline{J}_9$ and $\underline{J}_{10}$ into (4.29) we obtain,

$$\Longrightarrow \quad \underline{J}_{10}\, \Delta\underline{Y}_2 = \underline{J}_8\, \Delta\underline{z}_1 + \underline{J}_9\, \underline{J}_2^{-1}\, \Delta\underline{z}_2 \tag{4.32}$$

$$\Longrightarrow \quad \Delta\underline{Y}_2 = \underline{J}_{10}^{-1}\, \underline{J}_8\, \Delta\underline{z}_1 + \underline{J}_{10}^{-1}\, \underline{J}_9\, \underline{J}_2^{-1}\, \Delta\underline{z}_2 \tag{4.33}$$

Finally, defining the matrices

$$\underline{B}_1^T = \underline{J}_{10}^{-1}\, \underline{J}_8$$

$$\underline{B}_2^T = \underline{J}_{10}^{-1}\, \underline{J}_9\, \underline{J}_2^{-1} \tag{4.34}$$

and substituting in (4.33) we obtain the desired form:

$$\Delta\underline{Y}_2 = \underline{B}_1^T\, \Delta\underline{z}_1 + \underline{B}_2^T\, \Delta\underline{z}_2 \tag{4.35}$$

Equation (4.35) is the end result of simple algebraic manipulations performed on the incremental load flow equations. If the incremental changes were designated to be deviations from the base case, then the incremental model would be identical to the linear approximation. Consequently, the $\underline{B}$ vectors given by (4.34) would be identical to the $\underline{B}$ vectors found in section 4.2.

At first glance one may not discern and appreciate the advantages of using the incremental equations to update $\underline{B}$ as opposed to recalculating $\underline{B}$. For this reason equation (4.34) will now be examined in depth.

A look at the dimensions involved might prove helpful; assume that the external system comprises of N_1 buses among which L buses are connected via tie-lines to internal system buses.Then we have the following dimensions:

$$\Longrightarrow \quad \underline{J}_1 : 2N_1 * 2N_1$$

$$\underline{J}_3 : 2L * 2N_1$$

$$\underline{J}_4 : 2L * (2N_2 - 1)$$

Moreover, suppose the internal system includes N_2 buses among which there are M boundary buses. Then we have the following dimensions:

$$\Longrightarrow \quad \underline{J}_2 : (2N_2 - 1) * (2N_2 - 1)$$

$$\underline{J}_5 : 2M * 2N_1$$

$$\underline{J}_6 : 2M * (2N_2 - 1)$$

$$\underline{A}_2 \; : \; (2N_2 - 1) * 2M$$

$$\underline{A}_1 \; : \; 2N_1 * 2L$$

Accordingly;

$$\underline{J}_8 \; : \; 2M * 2N_1$$

$$\underline{J}_7 \, , \, \underline{J}_9 \; : \; 2M * (2N_2 - 1)$$

$$\underline{J}_{10} \; : \; 2M * 2M$$

$$\underline{B}_1 \; : \; 2M * 2N_1$$

$$\underline{B}_2 \; : \; 2M * (2N_2 - 1)$$

Usually the external system is larger than the internal system; $N_1 > N_2$. Moreover, there are normally very few boundary buses; $N_1 >> L$ and $N_2 >> M$.

The Jacobians $\underline{J}_1$,$\underline{J}_2$,$\underline{J}_3$,$\underline{J}_4$,$\underline{J}_5$, and $\underline{J}_6$ are very sparse matrices. Furthermore, matrix $\underline{A}_1$ contains at most 2L ´-1´ elements, and matrix $\underline{A}_2$ contains at most 2M ´+1´ elements. Thus matrices $\underline{A}_1$ and $\underline{A}_2$ are also very sparse.

A change in the internal network topology will incur changes in $\underline{J}_2$ only [i.e, all of $\underline{J}_1$, $\underline{J}_3$, $\underline{J}_4$, $\underline{J}_5$, $\underline{J}_6$, $\underline{J}_7$, $\underline{J}_8$, $\underline{J}_9$, $\underline{A}_1$ and $\underline{A}_2$ need to be calculated only once].

In order to approximate the tie-line flows, one may store the matrices $\underline{J}_9$, $\underline{A}_2$, $\underline{J}_8$ and use them along with the internal system Jacobian $\underline{J}_2$ to solve for $\underline{B}$ in (4.34) and subsequently substitute in (4.35).

The above method would compute matrix inverses and store the large and full matrix $\underline{J}_9$. A more computationally efficient approach is the following method. $\underline{J}_8$ is found by factorizing the sparse matrix $(\underline{J}_1 + \underline{A}_1 \underline{J}_3)$ and applying 2M forward and backward substitutions (recall that M is the number of internal boundary buses which are usually very few). The matrix $\underline{J}_8$ as well as the very sparse matrices $\underline{J}_4$, $\underline{J}_6$, $\underline{A}_1$ are stored. In order to determine the approximate tie-line flows it is necessary to compute $\underline{J}_7$ ($= \underline{J}_8 \underline{A}_1 \underline{J}_4$). This is not a demanding computation since $\underline{A}_1$ and $\underline{J}_4$ are very sparse. Next, $\underline{J}_9$ is found, as in (4.30). $\underline{J}_{10}$ is found, as in

(4.31), by first factorizing $\underline{J}_2$ and applying 2M forward and backward substitutions to evaluate $\underline{J}_2^{-1} \underline{A}_2$ (since $\underline{A}_2$ is very sparse). It is not computationally efficient to execute the intermediary step of finding the $\underline{B}$ vectors from (4.34). The approximations to the tie-line flows, $\Delta\underline{Y}_2$ should be found directly, by solving (4.32) using LU decomposition.

It is not necessary to recompute the factors of the internal system Jacobian every time a change in the internal system topology is studied. The new solution may be updated with the old factors either through the Matrix Inversion Lemma or via compensation techniques [26]. Accordingly, both $\underline{J}_{10}$ and the approximations to the tie-line flows may also be updated.

4.5.2 METHOD 2

Consider the power system shown in fig 10, which has been divided into internal and external systems. The boundary buses, which are the only buses from which tie-lines interconnecting the internal and external systems emanate, have been grouped separately.

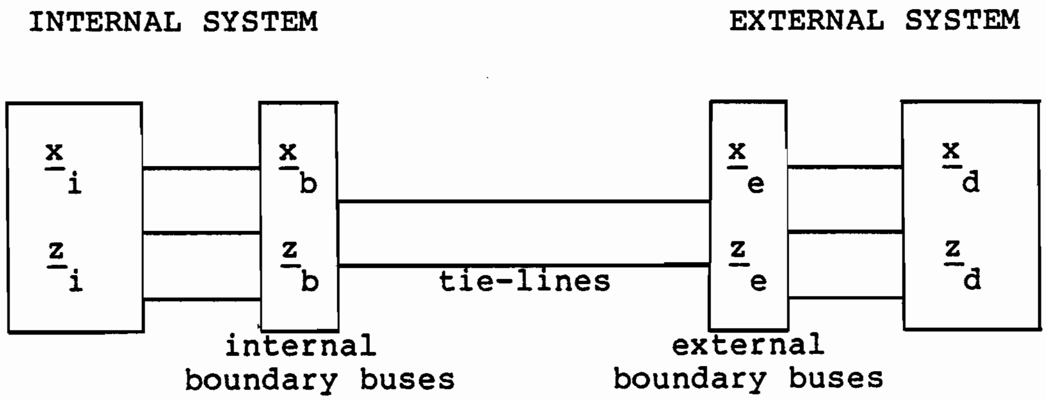


FIG 10

The nomenclature identifying each of the subsystems is specified in fig 10. Thus, the load flow equations for the complete system are:

$$\underline{z}_d = \underline{F}_d(\underline{x}_e, \underline{x}_d) \tag{4.36}$$

$$\underline{z}_e = \underline{F}_e(\underline{x}_b, \underline{x}_e, \underline{x}_d) \tag{4.37}$$

$$\underline{z}_b = \underline{F}_b(\underline{x}_i, \underline{x}_b, \underline{x}_e) \tag{4.38}$$

$$\underline{z}_i = \underline{F}_i(\underline{x}_i, \underline{x}_b) \tag{4.39}$$

Linearizing about the base case, we get;

$$
\begin{bmatrix} \Delta \underline{z}_d \\ \Delta \underline{z}_e \\ \Delta \underline{z}_b \\ \Delta \underline{z}_i \end{bmatrix}
=
\begin{bmatrix} \underline{J}_{dd} & \underline{J}_{de} & 0 & 0 \\ \underline{J}_{ed} & \underline{J}_{ee} & \underline{J}_{eb} & 0 \\ 0 & \underline{J}_{be} & \underline{J}_{bb} & \underline{J}_{bi} \\ 0 & 0 & \underline{J}_{ib} & \underline{J}_{ii} \end{bmatrix}
\begin{bmatrix} \Delta \underline{x}_d \\ \Delta \underline{x}_e \\ \Delta \underline{x}_b \\ \Delta \underline{x}_i \end{bmatrix}
\qquad (4.40)
$$

where $\underline{J}_{lk}$ is the partial derivative of $\underline{F}_l$ with respect to $\underline{x}_k$.

It is desired to obtain the reduced system shown in fig 11.

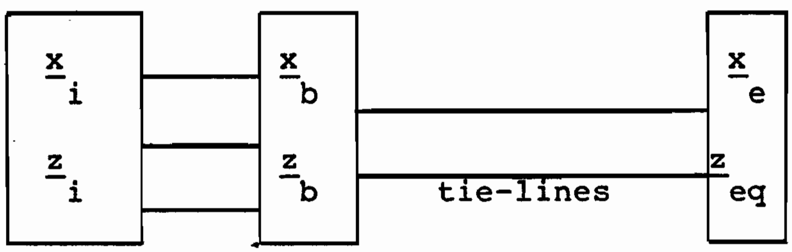


FIG 11

Fig 11 may be obtained from fig 10 by deleting all external buses (but not external boundary buses), there will be new external boundary bus interconnections as well as new external boundary bus injections. Let us now examine the effect of eliminating vector $\Delta \underline{x}_d$ in (4.40) using gaussian elimination:

$$\begin{bmatrix} \Delta \underline{z}_{eq} \\ \Delta \underline{z}_b \\ \Delta \underline{z}_i \end{bmatrix} = \begin{bmatrix} \underline{J}_{ee\ eq} & \underline{J}_{eb} & 0 \\ \underline{J}_{be} & \underline{J}_{bb} & \underline{J}_{bi} \\ 0 & \underline{J}_{ib} & \underline{J}_{ii} \end{bmatrix} \begin{bmatrix} \Delta \underline{x}_e \\ \Delta \underline{x}_b \\ \Delta \underline{x}_i \end{bmatrix} \qquad (4.41)$$

where

$$\underline{J}_{ee\ eq} = \underline{J}_{ee} - \underline{J}_{ed}\, \underline{J}_{dd}^{-1}\, \underline{J}_{de}$$

$$\Delta \underline{z}_{e\ eq} = \Delta \underline{z}_e - \underline{J}_{ed}\, \underline{J}_{dd}^{-1}\, \Delta \underline{z}_d \qquad (4.42)$$

The set of equations in (4.41) may be expressed in a more concise notation as :

$$\Delta \hat{\underline{z}} = [\, 2 \;\; \hat{\underline{L}}(\hat{\underline{x}}) \,]\, \Delta \hat{\underline{x}} \qquad (4.43)$$

where $\hat{\underline{L}}(\hat{\underline{x}})$ is one-half the jacobian of (4.41).

The Jacobian terms depend on the network topology and the voltages. Since elimination has only altered $\underline{J}_{ee}$,one may conclude that only the external system's boundary bus interconnections will be modified. Fortunately, we will not be interested in knowing explicitly what these changes are. To summarize, the elimination of $\underline{\Delta x}_d$ in (3.40) by gaussian elimination produces the network of fig 11.

Let us now consider how method 2 may be used to determine the desired approximations, assuming that $\underline{\Delta z}_{eq}$ and $\underline{J}_{ee\ eq}$ (a square matrix of dimension 2L, where L is the number of external boundary buses and is usually very small) have already been calculated and stored. All of the Jacobian matrices in (4.41) should be evaluated using the base case voltages. The linear approximations to each of the real and reactive tie-line powers flowing into the internal system's boundary buses are then found by using the approximation formula:

$$y = \underline{\hat{x}} \quad \underline{\hat{R}} \quad (\underline{\hat{L}}(\underline{\hat{x}}))^{-1} \quad \underline{\hat{z}} \qquad (4.44)$$

where y is the approximation to the real or reactive tie-line flow. The $\underline{\hat{R}}$ matrices for the real and reactive power flow approximations are shown in Appendix B. An interesting observation is that $\underline{\hat{R}}$ depends only on the topology of the tie-lines. Thus, it is irrelevant to know explicitly what the new external boundary interconnections are. One may then proceed, exactly as in section 4.2, by supplementing the internal system's boundary injections and subsequently implementing the reduced system load flow.

Method 2 still retains the property of not affecting the retained system Jacobian, therefore allowing the use of normal load flow programs.

CHAPTER 5

TEST CASES

Exhaustive testing of the proposed equivalencing method was performed using the 5 bus system shown in fig 8 and whose line data is provided in table 1. A selection of results that are typical of the cases treated are presented in the first section of this chapter. The next section is devoted to results obtained while using the IEEE 30 bus system. Finally, section 5.3 will comment on the numerical results obtained by using various equivalencing methods.

5.1 THE 5 BUS SYTEM

5.1.1 CLASSIFYING THE VARIOUS CASES

The injections specified in table 2 are designated as the base case injections, thus the corresponding load flow, given by table 3, is the base case load flow.

Specified load flow independent variables include:

$|V_1|$, $|V_2|$, $|V_3|$, P_{G2}, P_{D2}, Q_{D2}, P_{G3}, P_{D3}, Q_{D3}, P_{D4}, Q_{D4}, P_{D5}, Q_{D5}

Accordingly, by altering the above quantities various test cases may be obtained.

Throughout these examples, the voltage magnitudes have been maintained at base case values.

The real and reactive demands are expressed as a percentage of the base case values. Thus a 20% increase from base case is interpreted as:

$P_{D2} = .240$ $Q_{D2} = .120$

$P_{D3} = .540$ $Q_{D3} = .180$

$P_{D4} = .480$ $Q_{D4} = .060$

$P_{D5} = .720$ $Q_{D5} = .120$

Likewise, the real power generations are also expressed as a percentage of their base case values.

5.1.2 APPROACH

For each case cosidered, the following are determined;

1. The complete system load flow.
2. The approximate tie-line flows.
3. The reduced system load flow, with boundary injections supplemented by the linear approximations.

For most of the cases treated, two additional reduction methods were implemented.

4. Instead of using the linear approximation to augment the boundary bus injections in the reduced model, the base case tie-line flows are used.
5. A Ward reduction, where both external loads and generators are transformed to current injections.

4. and 5. were obtained so as to gain some insight into the behaviour of the approximation method. We are well aware that method 4 is a very crude and primitive method, and that more accurate results may be obtained from a different formulation of the Ward method.

5.1.3 Notation

linear	linear approximation.
quadratic	quadratic approximation.
exact	results from complete system load flow.
fixed	results obtained by method 4.

Ward	results obtained by method 5.
updated	approximation updated due to contingency.

In each case presented, the exact tie-line flows are compared to the corresponding linear and quadratic approximations. Next, results obtained by implementing the various methods are presented.

5.1.4 sample results

Case 1 — 5 % increase from base case conditions

no contingency

tie-line flow	Exact P +j Q	Linear P +j Q	Quadratic P +j Q
2-4	.164 -j .000	.164 -j .000	.164 -j .000
2-5	.469 +j .050	.469 +j .050	.469 +j .050
3-4	.329 -j .023	.329 -j .023	.329 -j .023

Dependent variables	Exact	Linear	Ward	Fixed
line flows				
1-2	.275+j.053	.275+j.053	.314+j.041	.313+j.041
1-3	.151+j.012	.151+j.012	.165+j.008	.164+j.008
2-3	.109+j.001	.109+j.001		.114-j.001
angles				
@ 2	-0.76	-0.76	-0.89	-0.89
@ 3	-1.71	-1.71	-1.90	-1.90

Case 2   20 % increase from base case conditions

no contingency

tie-line flow	Exact P +j Q	Linear P +j Q	Quadratic P +j Q
2-4	.207 -j .003	.207 -j .003	.207 -j .003
2-5	.594 +j .078	.594 +j .077	.594 +j .078
3-4	.417 +j .011	.417 +j .010	.417 +j .011

Dependent variables	Exact	Linear	Ward	Fixed
line flows				
1-2	.351+j.029	.351+j.029	.197+j.078	.195+j.079
1-3	.191+j.000	.191+j.000	.138+j.016	.138+j.016
2-3	.137-j.008	.137-j.008		.119-j.002
angles				
@ 2	-1.02	-1.02	-0.49	-0.49
@ 3	-2.27	-2.27	-1.53	-1.55

Case 3 40 % increase from base case conditions

no contingency

tie-line flow	Exact P +j Q	Linear P +j Q	Quadratic P +j Q
2-4	.242 -j .005	.242 -j .005	.242 -j .005
2-5	.695 +j .103	.694 +j .099	.695 +j .103
3-4	.488 +j .039	.487 +j .037	.488 +j .039

Dependent variables	Exact	Linear	Ward	Fixed
line flows				
1-2	.412+j.010	.410+j.010	.102+j.109	.100+j.110
1-3	.224-j.009	.224-j.009	.116+j.023	.118+j.022
2-3	.160-j.015	.160-j.015		.123-j.004
angles				
@ 2	-1.23	-1.22	-0.17	-0.16
@ 3	-2.72	-2.71	-1.24	-1.26

Case 4 60 % increase from base case conditions

no contingency

tie-line flow	Exact P +j Q	Linear P +j Q	Quadratic P +j Q
2-4	.278 -j .006	.277 -j .008	.278 -j .006
2-5	.797 +j .131	.794 +j .121	.797 +j .131
3-4	.559 +j .068	.557 +j .064	.559 +j .068

Dependent variables	Exact	Linear	Ward	Fixed
line flows				
1-2	.476-j.010	.472-j.008	.010+j.140	.006+j.141
1-3	.258-j.019	.257-j.018	.095+j.029	.098+j.028
2-3	.184-j.022	.184-j.022		.128-j.005
angles				
@ 2	-1.45	-1.43	0.15	0.16
@ 3	-3.19	-3.17	-0.95	-0.99

Case 5 80 % increase from base case conditions

no contingency

tie-line flow	Exact P +j Q	Linear P +j Q	Quadratic P +j Q
2-4	.313 -j .007	.312 -j .010	.313 -j .007
2-5	.900 +j .161	.894 +j .143	.899 +j .160
3-4	.631 +j .099	.628 +j .091	.631 +j .098

Dependent variables	Exact	Linear	Ward	Fixed
line flows				
1-2	.540-j.029	.532-j.027	-.083+j.171	-.088+j.173
1-3	.292-j.028	.290-j.027	.074+j.036	.078+j.035
2-3	.208-j.029	.207-j.029		.132-j.006
angles				
@ 2	-1.66	-1.64	0.47	0.49
@ 3	-3.65	-3.62	-0.65	-0.71

Case 6 100 % increase from base case conditions

no contingency

tie-line flow	Exact P +j Q	Linear P +j Q	Quadratic P +j Q
2-4	.349 -j .008	.347 -j .012	.349 -j .008
2-5	1.003 +j .194	.995 +j .164	1.003 +j .192
3-4	.702 +j .130	.698 +j .118	.702 +j .129

Dependent variables	Exact	Linear	Fixed
line flows			
1-2	.606-j.049	.594-j.045	.181+j.205
1-3	.327-j.037	.324-j.036	.058+j.041
2-3	.232-j.036	.231-j.036	.136-j.008
angles			
@ 2	-1.89	-1.85	0.81
@ 3	-4.12	-4.07	-0.44

Case 7   5 % decrease from base case conditions

no contingency

tie-line flow	Exact P +j Q	Linear P +j Q	Quadratic P +j Q
2-4	.181 -j .001	.181 -j .001	.181 -j .001
2-5	.519 +j .061	.519 +j .061	.519 +j .061
3-4	.364 -j .009	.364 -j .009	.364 -j .009

Dependent variables	Exact	Linear	Ward	Fixed
line flows				
1-2	.306+j.043	.306+j.043	.267+j.055	.266+j.056
1-3	.167+j.007	.167+j.007	.154+j.011	.154+j.011
2-3	.120-j.003	.120-j.003		.116-j.001
angles				
@ 2	-0.86	-0.86	-0.73	-0.73
@ 3	-1.94	-1.94	-1.75	-1.76

Case 8 20 % decrease from base case conditions

no contingency

tie-line flow	Exact P +j Q	Linear P +j Q	Quadratic P +j Q
2-4	.138 +j .002	.138 +j .002	.138 +j .002
2-5	.394 +j .034	.393 +j .033	.394 +j .034
3-4	.277 -j .043	.277 -j .043	.277 -j .043

Dependent variables	Exact	Linear	Ward	Fixed
line flows				
1-2	.229+j.068	.234+j.066	.383+j.019	.383-j.019
1-3	.126+j.020	.127+j.019	.180+j.003	.178+j.004
2-3	.091+j.007	.090+j.007		.110+j.001
angles				
@ 2	-0.60	-0.62	-1.13	-1.13
@ 3	-1.37	-1.38	-2.11	-2.09

Case 9 40 % decrease from base case conditions

no contingency

tie-line flow	Exact P +j Q	Linear P +j Q	Quadratic P +j Q
2-4	.103 +j .004	.103 +j .004	.103 +j .004
2-5	.294 +j .016	.293 +j .012	.294 +j .016
3-4	.207 -j .068	.207 -j .070	.207 -j .068

Dependent variables	Exact	Linear	Ward	Fixed
line flows				
1-2	.171+j.086	.170+j.087	.478-j.010	.479-j.011
1-3	.094+j.029	.094+j.030	.202-j.003	.199-j.002
2-3	.068+j.014	.068+j.014		.105+j.002
angles				
@ 2	-0.40	-0.40	-1.45	-1.46
@ 3	-0.93	-0.93	-2.41	-2.38

Case 10 60 % decrease from base case conditions

no contingency

tie-line flow	Exact P +j Q	Linear P +j Q	Quadratic P +j Q
2-4	.069 +j .008	.068 +j .006	.069 +j .008
2-5	.196 -j .001	.193 -j .010	.196 -j .000
3-4	.138 -j .093	.136 -j .097	.138 -j .093

Dependent variables	Exact	Linear	Ward	Fixed
line flows				
1-2	.113+j.105	.109+j.107	.572-j.039	.575-j.040
1-3	.062+j.040	.061+j.040	.223-j.009	.219-j.008
2-3	.045+j.021	.045+j.021		.101+j.003
angles				
@ 2	-0.21	-0.19	-1.77	-1.78
@ 3	-0.50	-0.48	-2.70	-2.65

Case 11 90 % decrease from base case conditions

no contingency

tie-line flow	Exact P +j Q	Linear P +j Q	Quadratic P +j Q
2-4	.017 +j .013	.016 +j .010	.017 +j .013
2-5	.049 -j .022	.043 -j .043	.049 -j .021
3-4	.035 -j .128	.031 -j .137	.035 -j .128

Dependent variables	Exact	Linear	Ward	Fixed
line flows				
1-2	.029+j.133	.020+j.136	.714-j.081	.719-j.082
1-3	.016+j.055	.013+j.056	.256-j.018	.250-j.017
2-3	.011+j.033	.011+j.033		.094+j.006
angles				
@ 2	0.08	0.11	-2.26	-2.27
@ 3	0.15	0.19	-3.15	-3.07

Case 12   20 % increase from base case conditions

contingency; line 2-3 removed

tie-line flow	Exact P +j Q	Linear P +j Q	Quadratic P +j Q
2-4	.269 -j .021	.269 -j .022	.269 -j .021
2-5	.625 +j .071	.624 +j .069	.625 +j .071
3-4	.327 +j .041	.326 +j .040	.327 +j .041

	Exact	Linear	Linear updated	Fixed
line flows				
1-2	.305+j.043	.212+j.073	.305+j.043	.075+j.118
1-3	.239-j.013	.333-j.038	.238-j.013	.260-j.019
angles				
@ 2	-0.86	-0.54	-0.86	-0.08
@ 3	-2.92	-4.20	-2.91	-3.21

Case 13 40 % increase from base case conditions

contingency; line 2-3 removed

tie-line flow	Exact P +j Q	Linear P +j Q	Quadratic P +j Q
2-4	.315 -j .025	.314 -j .027	.315 -j .025
2-5	.731 +j .096	.729 +j .090	.731 +j .096
3-4	.383 +j .074	.382 +j .072	.382 +j .074

	Exact	Linear	Linear updated	Ward	Fixed
line flows					
1-2	.359+j.026	.248-j.061	.356+j.027	.063+j.122	-.023+j.151
1-3	.280-j.025	.390-j.052	.280-j.025	.157+j.010	.243-j.015
angles					
@ 2	-1.05	-0.67	-1.04	-0.03	0.26
@ 3	-3.48	-4.97	-3.47	-1.79	-2.98

Case 14 80 % decrease from base case conditions

contingency; line 2-3 removed

tie-line flow	Exact P +j Q	Linear P +j Q	Quadratic P +j Q
2-4	.407 -j .032	.404 -j .038	.407 -j .032
2-5	.946 +j .153	.940 +j .131	.946 +j .152
3-4	.495 +j .144	.492 +j .135	.494 +j .144

	Exact	Linear	Linear updated	Ward	Fixed
line flows					
1-2	.471-j.008	.332+j.038	.462-j.005	-.124+j.185	-.219+j.218
1-3	.365-j.046	.508-j.079	.362-j.046	.116+j.023	.211-j.006
angles					
@ 2	-1.43	-0.92	-1.40	.61	0.94
@ 3	-4.63	-6.56	-4.60	-1.23	-2.54

Case 15   40 % decrease from base case conditions

contingency; line 2-3 removed

tie-line flow	Exact P +j Q	Linear P +j Q	Quadratic P +j Q
2-4	.133 -j .005	.133 -j .005	.133 -j .005
2-5	.309 +j .012	.308 +j .008	.309 +j .012
3-4	.162 -j .054	.162 -j .056	.162 -j .053

	Exact	Linear	Linear updated	Ward	Fixed
line flows					
1-2	.148+j.094	.102+j.109	.147+j.094	.441+j.001	.372+j.022
1-3	.117+j.022	.163+j.008	.117+j.022	.239-j.014	.308-j.032
angles					
@ 2	-0.33	-0.17	-0.32	-1.33	-1.09
@ 3	-1.25	-1.88	-1.25	-2.92	-3.86

Case 16 — 90 % increase from base case conditions

contingency; line 2-3 removed

tie-line flow	Exact P +j Q	Linear P +j Q	Quadratic P +j Q
2-4	.022 +j .011	.020 +j .008	.022 +j .011
2-5	.051 -j .022	.045 -j .044	.051 -j .021
3-4	.027 -j .126	.024 -j .135	.027 -j .125

	Exact	Linear	Linear updated	Ward	Fixed
line flows					
1-2	.025+j.135	.010+j.140	.016+j.137	.723-j.083	.623-j.054
1-3	.019+j.054	.023+j.052	.017+j.055	.316-j.034	.348-j.042
angles					
@ 2	0.10	0.15	0.13	-2.28	-1.95
@ 3	0.10	0.04	0.14	-3.96	-4.40

5.2 THE IEEE 30 BUS SYSTEM

In his Phd thesis [10], E.H. Elkonyaly cited results rendered by various equivalencing methods while using the IEEE 30 bus system. This section will investigate the same contingencies and compare the performance of the linear approximation equivalent to the corresponding results as given in [10]. The IEEE 30 bus data used in this study was obtained from [25].

The internal (retained) buses are : 1, 2, 3, 4, 5, 6, 7, 8 and 28 - with buses 4, 6 and 28 being boundary buses. Injections are maintained at their base case values, the following contingencies are simulated :

1. Outage of line 1-2 (one circuit).
2. Outage of line 1-2 (both circuits).
3. Simultaneous outages of lines 2-4 and 2-6.
4. Simultaneous outages of lines 3-4, 5-7, 6-8.

For every contingency discussed, the exact tie-line flows and the linear approximations to the tie-line flows (as given by the approximation program) are given. Next, the results of implementing various equivalencing techniques are presented; namely Ward Classical (Y) (using equivalent admittances to model loads <u>and</u> equivalent current injections to model generators), Ward Classical (I) (using equivalent current injections to model both loads and generators), REI, Linearized Jacobian, and the Linear Approximation. As a means of comparison two error criteria are used. The first one consists of evaluating the maximum difference between the exact contingency voltage magnitude or angle and each of the reduction techniques voltage magnitude or angle for all nodes within the retained network. The second error criteria used is the sum of the absolute values of these discrepencies.

1. Outage of line 1-2 (one circuit)

boundary line flows	actual	linear approximation
4-12	.24789 +j.00126	.24788 +j.00136
6-9	.13266 -j.03847	.13267 -j.03844
6-10	.11444 +j.00420	.11444 +j.00421
28-27	.16428 +j.05813	.16428 +j.05813

reduction technique	maximum error		Σ |error|	
	|E|	@	E	@
Classical(Y)	1.7E-1	17.9	5.4E-1	101.6
Classical(I)	3.6E-3	6.7E-2	1.3E-2	2.2E-1
REI	.2E-3	.5E-1	.3E-3	.2
Linearized	6.5E-5	2.1E-3	2.0E-4	8.3E-3
Approximation	1.0E-5	5.1E-5	1.0E-5	2.8E-4

2. Outage of line 1-2 (both circuits)

boundary line flows	actual	linear approximation
4-12	.26439 -j.00463	.26373 +j.00042
6-9	.12410 -j.03868	.12437 -j.03720
6-10	.10957 +j.00458	.10975 +j.00493
28-27	.16156 +j.05918	.16170 +j.05909

reduction technique	maximum error		Σ |error|	
	|E|	@	E	@
Classical(Y)	3.6E-4	4.7E-2	1.4E-3	3.2E-1
Classical(I)	5.7E-4	8.0E-3	2.4E-3	2.8E-2
REI	.3E-2	1.1	.9E-2	7.5
Linearized	3.4E-6	1.9E-4	1.2E-5	9.3E-4
Approximation	2.2E-4	4.6E-3	6.7E-4	2.4E-2

3. Simultaneous outages of lines 2-4 and 2-6

boundary line flows	actual	linear approximation
4-12	.25368 -j.02830	.25332 -j.02539
6-9	.13027 -j.05472	.13040 -j.05346
6-10	.11282 -j.00072	.11292 -j.00036
28-27	.16278 +j.05783	.16288 +j.05783

reduction technique	maximum error		Σ |error|	
	|E|	@	E	@
Classical(Y)	1.8E-3	1.4	7.6E-3	10.3
Classical(I)	7.4E-3	6.1E-1	3.2E-2	3.8
REI	.7E-3	.3	.2E-2	1.4
Linearized	4.5E-4	5.8E-2	1.7E-3	4.0E-1
Approximation	1.9E-4	2.03E-3	5.0E-4	5.0E-3

4. Simultaneous outages of lines 3-4, 5-7 and 6-8

boundary line flows	actual	linear approximation
4-12	.23823 -j.01865	.23812 -j.01747
6-9	.14124 -j.04779	.14120 -j.04668
6-10	.11916 +j.00123	.11915 +j.00159
28-27	.16064 +j.05869	.16077 +j.05882

reduction technique	maximum error		Σ|error|	
	|E|	@	E	@
Classical(Y)	7.5E-3	5.5E-1	3.0E-2	3.7
Classical(I)	1.4E-2	4.2E-1	5.9E-2	8.0E-1
REI	.1E-2	.4	.4E-2	1.8
Linearized	2.2E-4	8.3E-3	8.7E-4	2.0E-2
Approximation	1.9E-4	2.8E-3	7.0E-4	1.0E-2

5.3 COMMENTS ON THE NUMERICAL RESULTS

When comparing the various equivalencing methods, one must keep in mind that the conditions being assumed are, by and large, of a theoretical nature. For instance, we may assume that all the pertinent external information is available, that there are no errors in the internal state estimator data and that the network is three phase balanced. In order to fully determine the behaviour of a particular method, the only viable alternative is to actually implement it on-line over a sufficiently long period of time. This is a very costly, if not impractical, approach; yet, even if it were feasible, one would still lack the tools necessary for establishing a meaningful comparison. More bizzare is the fact that equivalents have been found to be highly problem dependent. To summarize, one must be cautious while interpreting the numerical results of equivalencing methods.

Glancing through the results of the 5 bus system, it is evident that the quadratic approximation of the tie-line flows is consistently very close to the actual flows. Unfortunately, the excessive computational effort required as well as the necessity to store large and full matrices diminish the appeal of the method for on-line applications.

Looking at the linear approximation results, one observes that the approximations to the real power flows are generally more accurate than their reactive counterparts. A 5 bus system, due to its compactness, tends to bring out the worst in an equivalent. Especially so for our method when a contingency is simulated, since the $\underline{B}$ vectors and the $\underline{C}$ matrices are essentially sensitivity elements and are thus likely to be highly sensitive to the few branches that are present. In a larger network the elements of $\underline{B}$ and $\underline{C}$ would be less sensitive to the individual branches, which will result in more accurate approximations.

Results of the IEEE 30 bus system seem to indicate that the Linear Approximation method is in the same league as the Linearized Jacobian method, which is quite satisfactory.

CHAPTER 6

CONCLUSIONS AND RECOMMENDATIONS

- The equivalencing method introduced in this thesis exhibits the properties desirable in a good equivalent, mainly;

  . The method does not introduce any additional buses to the internal system.

  . The sensitivity elements (of $\underline{B}$, $\underline{C}$) provide a readily identifiable relationship between the external system generations and the additional boundary injections.

  . The equivalent may be used for interchange studies.

  . It is possible to adjust the equivalent so that it would be valid over a wide range of operating conditions.

- Of particular significance is the fact that the internal network topology is not changed by the proposed approach. Thus, the load flow programs that are currently available in the industry may be used without modifications.

- Two methods for updating the linear approximations to the tie-line flows at the internal boundary buses were presented. Method 1 found an explicit expression for evaluating the updated approximations by linearizing the load flow equations about a known base case. The only variable matrix in the expression was the internal system Jacobian. Method 2 also commenced by linearizing the load flow equations about a known base case. Gaussian elimination was then performed on the linearized equations to reduce the effect of the external system to the external system boundary buses. The tie-line flows (or $\underline{B}$ vectors) were then found by using the linear approximation formula associated with the reduced network. In theory, both Method 1 and Method 2 should give identical numerical results. Method 2 appears to involve less storage than Method 1. Furthermore, since Method 2 requires only one factorization (of the order of the internal system plus external system boundary buses) and several multiplications with the sparse $\underline{R}$ matrices, it might also be more computationally efficient.

- The quadratic approximations to the tie-line flows were found to be consistently very close to the actual flows. Unfortunately, the excessive computational effort required as well as the necessity to store large and full matrices diminish the appeal of the quadratic approximation for on-line implementation. It remains to be seen whether there exists an efficient method for simply updating the approximation.

- The performance of the equivalent, which uses the linear approximations to the tie-line flows as boundary injections, was found to be as good as the Linearized Jacobian equivalencing method, and seems to be more accurate than the Ward and the REI equivalents (based on the test cases using the IEEE 30 bus test system). The results quoted in reference [17] confirm the high performance of linearized methods as compared to other commonly used equivalencing methods.

- In this work, the approximate tie-line flows were found before implementing the reduced system load flow. It is possible to insert the approximation program into the load flow program, so that at each iteration the Jacobian of the approximation program is evaluated at the updated internal voltages. This will result in more accurate approximations, but it remains to be seen whether the additional computational effort required is justified.

- An interesting point that is worth investigating concerns the updating of the $\underline{B}$ vector. Suppose that $\underline{B}_1$ is the precontingency sensitivity vector, and $\underline{B}_2$ is the post-contingency vector. It is possible to find a vector $\underline{\Delta B}$, such that $\underline{B}_2 = \underline{B}_1 + \underline{\Delta B}$, using compensation techniques [26].

A tremendous saving in the computational effort that was previously required might follow if this method for updating $\underline{B}$ were implemented.

REFERENCES

[1] J.B. Ward, "Equivalent Circuits for Power-Flow Studies", AIEE Trans., Vol. 68, pp. 373-382, 1949.

[2] H. Duran, N.V. Arvanitidis, "Simplifications for Area Security Analysis: A New Look at Equivalence", IEEE Trans., PAS. Vol.91 , pp. 670-679, 1972.

[3] E.K. Paulsson, "Network Equivalents for On-Line Systems", Paper C 74 373-7, IEEE PES Summer Meeting, 1974.

[4] A.S. Debs, "Estimation of External Network Equivalents from Internal System Data", IEEE Trans., PAS. Vol. 94, pp. 273-279, 1975.

[5] J.F. Dopazo, M.H. Dwarakanath, J.J. Li, A.M. Sasson, "An External System Equivalent Model Using Real-Time Measurements for System Security Evaluation", IEEE Trans., Vol. 96, pp. 431-446, 1977.

[6] G. Contaxis, A.S. Debs, "Network Equivalents for On-Line Power System Security Assessment",IEEE

Southeastern Conference and Exhibit, 1976.

[7] G. Contaxis, A.S. Debs, "Identification of External System Equivalents for Steady-State Security Assessment", IEEE Trans., PAS Vol 97, pp. 409-414, 1978.

[8] F.L. Alvarado, E.H. Elkonyaly, "Reduction in Power Systems", Paper A 77 507-7, IEEE PES Summer Meeting, 1977.

[9] F.L. Alvarado, E.H. Elkonyaly, "External System Static Equivalent for On-Line Implementation", Paper A 78 060-6, IEEE Pes Summer Meeting, 1978.

[10] E. H. Elkonyaly, "Development of a Static Equivalent for On-Line Security Analysis of Large-Scale Power Systems", Ph.D. Thesis, University of Wisconsin-Madison, 1977.

[11] W.F. Tinney, W.L. Powell, "The REI Approach to Power Network Equivalents", PICA Conference, pp. 314-320, 1977.

[12] T.E. Dy Liacco, S.C. Savulescu, K.A. Ramarao, "An On-Line Topological Equivalent of a Power Sytem",

Paper F 77 523-4, IEEE PES Summer Meeting, 1977.

[13] A. Monticelli, S. Deckmann, A. Garcia, B. Stott, "Real-Time External Equivalents for Static Security Analysis", IEEE TRANS., PAS VOL. 98, PP. 498-508, 1979.

[14] J.F. Dopazo, G. Irisarri, A.M. Sasson, "Real-Time External System Equivalent for On-Line Contingency Analysis", IEEE Trans., PAS Vol. 98, pp. 2153-2171, 1979.

[15] F. Wu, N. Narasimhamurthi, "Necessary Conditions for REI Reduction to be Exact", Paper A 79 065-4,IEEE PES Winter Meeting, 1979.

[16] E.C. Housos, G. Irisarri, R.M. Porter, A.M. Sasson, "Steady State Network Equivalents for Power System Planning Applications", PAS Vol. 99, pp. 2113-2120, 1980.

[17] S. Deckmann, A. Pizzolante, A. Monticelli, B. Stott, O. Alsac, "Numerical Testing of Power System Load Flow Equivalents", PAS Vol. 99, pp. 2292-2300, 1980.

[18] S. Deckmann, A. Pizzolante, A. Monticelli, B. Stott, O. Alsac, "Studies on Power System Load Flow Equivalencing", PAS Vol. 99, pp. 2301-2310, 1980.

[19] IEEE Commitee Report, "External Information Requirements for Power System Operations", Paper A 79 119-9, IEEE PES Winter Meeting, 1978.

[20] F.D. Galiana, "Power Voltage Limitations Imposed by the Network Structure of a Power System", PICA Conference, pp. 356-365, 1975

[21] F.D. Galiana, K. Lee, "On the Steady State Stability of Power Systems", PICA Conference, pp. 201-210, 1977.

[22] J. Jarjis, "Loadflow Feasibility", Ph.D. Thesis, McGill University, 1980.

[23] F.D. Galiana, M. Banakar, "Approximation Formulae for Dependent Load Flow Variables", Paper A 79 078-7, IEEE PES Winter Meeting, 1979.

[24] S.L. Low, "The Load Flow Problem Without Slack Bus", Masters Thesis, McGill University, 1980.

[25] B. Stott, O. Alsac, "Optimal Load Flow with Steady-State Security", IEEE Trans., PAS Vol. 93, pp. 745-751, 1974.

[26] F.D. Galiana, H. Glavitch, A. Fiechter, "A General Compensation Method for the Study of Line Outages in Load Flow Problems", Power Systems Computation Conference, 1975.

[27] G.W. Stagg, A.H. El-Abiad, "Computer Methods in Power Sytem Analysis",New York: McGraw-Hill, 1968.

[28] P. Dimo, "Nodal Analysis of Power systems",England: Abacus Press, 1975.

[29] J. Brameller, "Practical Diakoptics for Electrical Networks", England:Chapman and Hall Ltd., 1969.

[30] U. Knight, "Power System Engineering and Mathematics", Germany:Peramon Press, 1972.

APPENDIX A

R MATRICES

A.1 FORMING THE R MATRIX CORRESPONDING TO TRANSMISSION LOSSES

Consider a branch of impedance Z ;

$$Z = R_{ik} + j\,X_{ik} = \frac{1}{G_{ik} + j\,B_{ik}} \qquad \text{(A.1)}$$

connecting bus i and bus k

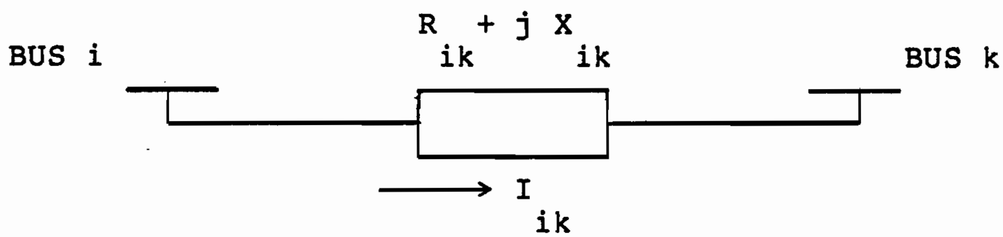


FIG 12

The real power loss in branch ik , fig 12, is given by

$$P_{L_{ik}} = G_{ik}\,V_{ik}^{2} \qquad \text{(A.2)}$$

where $|V|$ denotes the voltage magnitude, and $V_{ik} = V_i - V_k$.

The total transmission loss is obtained by the following summation

$$P_L = \frac{1}{2} \sum_{i=1}^{N} \sum_{i=1}^{N} V_{ik}^2 \; G_{ik} \qquad (A.3)$$

Substituting $V_{ik} = e_{ik} + j\, f_{ik}$ and expanding ;

$$P_L = \frac{1}{2} \sum_{i=1}^{N} \sum_{k=1}^{N} G_{ik} \left(e_i^2 - 2\, e_i\, e_k + e_k^2 \right)$$

$$+ \frac{1}{2} \sum_{i=1}^{N} \sum_{k=1}^{N} G_{ik} \left(f_i^2 - 2\, f_i\, f_k + f_k^2 \right) \qquad (A.4)$$

or more compactly written

$$P_L = \underline{e}^T \underline{G}\, \underline{e} + \underline{f}^T \underline{G}\, \underline{f} \qquad (A.5)$$

$$\Longrightarrow P_L = (\underline{e}^T \quad \underline{f}^T) \begin{bmatrix} \underline{G} & 0 \\ 0 & \underline{G} \end{bmatrix} \begin{bmatrix} \underline{e} \\ \underline{f} \end{bmatrix} \qquad (A.6)$$

$$\Longrightarrow \qquad P_L = \underline{X}^T \underline{R}\, \underline{X} \qquad (A.7)$$

where $\underline{G}$ is the real part of the admittance matrix.

Therefore

$$\underline{R} = \begin{bmatrix} \underline{G} & 0 \\ 0 & \underline{G} \end{bmatrix} \tag{A.8}$$

A.2 OTHER R MATRICES

The $\underline{R}$ matrix corresponding to the real power injections at bus k is :

$$p_k = \underline{x}^T \underline{P}_k \underline{x} \tag{A.9}$$

with

$$\underline{R} = \underline{P}_k = \begin{bmatrix} \underline{G}\,\underline{H}_k + \underline{H}_k\underline{G} & \underline{B}\,\underline{H}_k - \underline{H}_k\underline{B} \\ -\underline{B}\,\underline{H}_k + \underline{H}_k\underline{B} & \underline{G}\,\underline{H}_k + \underline{H}_k\underline{G} \end{bmatrix} \tag{A.10}$$

The $\underline{R}$ matrix corresponding to the reactive power injection at bus k is :

$$q_k = \underline{x}^T \underline{Q}_k \underline{x} \tag{A.11}$$

with

$$\underline{R} = \underline{Q}_k = \begin{bmatrix} \underline{B}\,\underline{H}_k + \underline{H}_k\,\underline{B} & -\underline{G}\,\underline{H}_k + \underline{H}_k\,\underline{G} \\ \underline{G}\,\underline{H}_k - \underline{H}_k\,\underline{G} & \underline{B}\,\underline{H}_k + \underline{H}_k\,\underline{B} \end{bmatrix} \tag{A.12}$$

Where $\underline{H}_k$ is an N*N matrix having all its entries zero except for the kk^{th} element which is 1/2 .

$\underline{P}_k$ and $\underline{Q}_k$ are 2N*2N symmetric and highly sparse matrices .

The $\underline{R}$ matrix for the square of the nodal voltage magnitude is :

$$|v_k|^2 = \underline{x}^T \underline{V}_k \underline{x} \tag{A.13}$$

$\underline{V}_k$ is obtained from $\underline{P}_k$ by replacing the matrices $\underline{G}$ and $\underline{B}$ by the unity matrix .

APPENDIX B

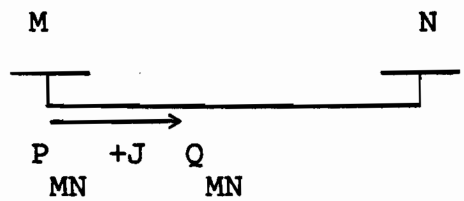
APPROXIMATION PROGRAM

```
$WATFIV
C
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CC                                                          CC
CC                                                          CC
CC                                                          CC
CC     THIS PROGRAM CALCULATES THE  B  VECTOR AND THE  C    CC
CC     MATRIX FOR THE LINEAR AND QUADRATIC APPROXIMATION    CC
CC     FORMULAE .                                           CC
CC                                                          CC
CC                                                          CC
CC                                                          CC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C
C
C
C          APPLICATION OF APPROXIMATE FORMULAE
C       ******************************************
C
C
C
C     FOR LOAD FLOW PURPOSES THERE ARE THREE TYPES OF BUSES
C
C      TYPE         NAME             SPECIFY          UNKNOWN
C
C      ____         _____            _______          _______
C
C       1            PQ              P , Q           V , ANGLE
C       2            PV              P , V           Q , ANGLE
C       3           SLACK            V ,ANGLE        P , Q
C
C
C
C   THE PURPOSE OF THE LINEAR AND QUADRATIC APPROXIMATIONS
C   IS TO RELATE AN ARBITRARY LOAD FLOW VARIABLE  Y  ,  TO
C   THE INDEPENDENT INJECTIONS OF A GENERAL LOAD FLOW
C   PROBLEM Z .
C
```

DEPENDENT VARIABLES Y , INCLUDE :

. $|V|^2$ AT EACH PQ BUS

. P_{MN} , Q_{MN}



. REAL LOSSES

. P GENERATION , Q GENERATION AT SLACK BUS

. Q GENERATION AT PV BUSES

GENERAL DESCRPTION OF ALGORITHM
****

1. COMPUTE JACOBIAN

2. FIND $L(X_0)$

   RECALL $L(X_0)$ = 1/2 JACOBIAN EVALUATED AT X_0

3. FIND R

4. COMPUTE $B^T = X_0^T \; R \; L^{-1}(X_0)$

5. FIND J(B)

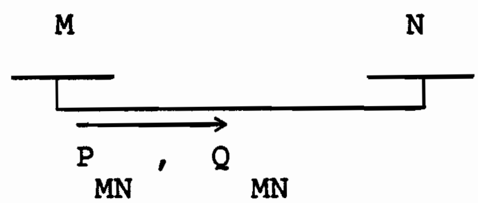
6. CALCULATE

$$C = 1/4 \; (L(X_0)^{T})^{-1} \; ((R)-J(B)) \; L(X_0)^{-1}$$

*** STEPS 4 AND 6 ARE FOUND USING LU DECOMPOSITION

```
C
C
C
C
C
C   THE  R  MATRIX
C  *****************
C
C
C
C
C   ** R  MATRIX FOR  P  GENERATION AT SLACK BUS
C
C
C
C
C                             ___                                    ___
C                            | G  H    + H   G       B  H   -  H   B  |
C                            |     K      K              K      K      |
C                            |                                         |
C         R = 1/2            |                                         |
C                            | -B H    + H   B       G  H   +  H   G  |
C                            |     K      K              K      K      |
C                            |___                                   ___|
C
C
C
C      WHERE  :     .  K = BUS NUMBER
C                   .  G = REAL  Y  ADMITTANCE
C                   .  B = IMAG  Y  ADMITTANCE
C                   .  H = HAS ZEROS EVERYWHERE EXCEPT THE
C                              KK SPOT THERE IS A "ONE" .
C
C
C
C     THE  R  MATRIX FOR  Q  GENERATION
C        AT SLACK AND/OR  PV BUSES
C                             ___                                  ___
C                            | -B  H  - H   B      G  H   -  H   G  |
C                            |      K    K             K      K      |
C       R  =  1/2            |                                       |
C                            |                                       |
C                            | -G  H  + H   G     -B  H   -  H   B  |
C                            |      K    K             K      K      |
C                            |___                                 ___|
C
C
C
C
C
C
```

THE R MATRIX FOR DETERMINING THE SQUARE OF THE VOLTAGE AT PQ BUSES
FOR BUS K: HAVE "0" EVERYWHERE EXCEPT FOR THE KK, (N+K)(N+K) ELEMENTS WHICH ARE EQUAL TO "1" .



THE R MATRIX
FOR REAL POWER TRANSFER

(** S __ # OF BUSES)

	M	N	M+S	N+S
M	.. $-G_{MN}$ $-1/2\ G_{MN}$ ..	$+1/2\ G_{MN}$ ..	0 ..	$-\ 1/2\ B_{MN}$ ..
N	.. $+1/2\ G_{MN}$ ..	0 ..	$+1/2\ B_{MN}$ ..	0 ..
M+S	.. 0 ..	$+1/2\ B_{MN}$ ..	$-G_{MN}$ $-1/2\ G_{MN}$ ..	$1/2\ G_{MN}$ ..
N+S	.. $-1/2\ B_{MN}$ ..	0 ..	$+1/2\ G_{MN}$ ..	0 ..

```
C         THE   R  MATRIX
C     FOR REACTIVE POWER TRANSFER
C
C              M             N           S+M          S+N
C
C
C    M  ..B   + 1/2 B  ..-1/2 B    ..    0        ..-1/2 G  ..
C          MN       MN        MN                         MN
C
C
C    N  ..-1/2 B       ..   0     .. 1/2 G       ..   0     ..
C              MN                      MN
C
C
C  M+S  ..    0        ..1/2 G    ..B   + 1/2 B  .. -1/2 B ..
C                            MN     MN       MN         MN
C
C
C  N+S  ..-1/2 G       ..   0     ..   -1/2 B    ..    0    ..
C              MN                          MN
C
C
C
C
C
C     ALL ELEMENTS NOT SHOWN ARE ZERO.
C
C
C
C
C
C      INPUT DATA
C     ************
C
C
C     1.    IN  INTEGER FREE FORMAT , ENTER THE TOTAL NUMBER
C           OF BUSES AND THE NUMBER OF  PV  BUSES .
C
C     2.    ENTER IN ORDER ON ONE CARD , USING INTEGER FREE
C           FORMAT
C              1  IF THE CORRESPONDING SELECTION IS DESIRED
C              0  OTHERWISE
C
C           * TO FIND SLACK BUS  P GEN
C
C           * TO FIND  Q GEN  AT ANY OF SLACK BUS OR PV BUS
C
C           * TO FIND  VSQRD  AT ONE OR MORE  PQ  BUSES
C
C           * TO FIND THE POWER TRANSFER TO ONE OR MORE BUSES
C
C           * TO FIND THE QUADRATIC APPROXIMATION TO THE
```

```
C           INDEPENDENT VARIABLE
C
C         * THIS PROGRAM HAS THE FLEXIBILTY TO ACCEPT
C            . AN INJECTION VECTOR
C            . TO CALCULATE THE INJECTION VECTOR FROM A
C              KNOWN VOLTAGE PROFILE
C           TO SELECT THE FIRST  PROCEDURE ENTER   1
C           TO SELECT THE SECOND PROCEDURE ENTER   0
C
C      3. IN  (F5.0)  FORMAT ENTER ON ONE CARD
C            1  IF THE CORRESPONDING SELECTION IS TO BE
C               PRINTED
C            0  OTHERWISE
C
C         * THE INPUT VOLTAGE PROFILE
C         * LINE DATA
C         * ADMITTANCE MATRIX
C         * PUNCH CARDS FOR  B  AND  C
C         * PRINT  B VECTOR AND  C MATRIX
C
C      4. EITHER THE INJECTION VECTOR , OR THE VOLTAGE PROFIL
C         VECTOR IS ENTERED AT THIS STAGE  (DIMENSION=2*N -1)
C         THE DATA MUST BE IN AGREEMENT WITH THE SELECTION
C         MADE IN STEP 2 .
C         ENTER THE DATA ON SEPARATE CARDS USING   (F10.5)
C         FORMAT
C         (IN  P.U.)
C
C         * THE CONVENTION USED IN THIS PROGRAM IS THAT :
C               .  SLACK BUS IS LABELED  # 1
C               .  PV BUSES ARE LABELED NEXT
C               .  PQ BUSES ARE LABELED LAST
C
C         * ORDER OF INJECTION VECTOR DATA
C               .  VOLTAGE SQUARED AT SLACK AND PV  BUSES
C               .  REACTIVE POWER AT PQ BUSES
C               .  REAL POWER AT  PV , THEN  AT  PQ BUSES
C
C         * ORDER OF VOLTAGE PROFIL VECTOR
C               .  REAL PART OF VOLTAGE FOR ALL BUSES
C               .  IMAGINARY PART OF VOLTAGE FOR ALL BUSES
C                  EXCEPT SLACK
C
C         IT IS POSSIBLE TO CONSIDER SEVERAL INJECTION
C         VECTORS , HOWEVER THE MATRIX ZINJ MUST BE
C         APPROPRIATELY DIMENSIONED . EXAMPLE ,FOR 3 INJECTIONS
C                      ZINJ(3,N2)
C         AT THE END OF THE PREVIOUS DATA INSERT A CARD ON
C         WHICH  999 IS ENTERED IN  (F10.5) FORMAT
C
C      INSERT A BLANK CARD
```

```
C
C       5. THE BASE VOLTAGES ARE ENTERED USING  (5F10.5) FORMAT  .
C          (IN  P.U.)
C
C       INSERT A BLANK CARD
C
C       6. EACH LINE DATA IS TYPED ON A SEPARATE CARD . ENTER
C          IN SEQUENCE , USING (2F5.0) FORMAT ;
C          NODE NUMBER.. JOINS NODE NUMBER
C          USING  (F10.5) FORMAT , ENTER ;  LINE RESISTANCE,
C          LINE REACTANCE , 1/2 CHARGING ADMITTANCE.
C          (IN P.U.)
C
C       INSERT A BLANK CARD
C
C       7. PV  BUS NUMBERS WHERE THE  Q  GENERATION IS TO BE
C          FOUND  . ENTER EACH BUS NUMBER ON A SEPARATE CARD
C          using (F5.0) FORMAT .
C
C       INSERT A BLANK CARD
C                                          2
C        8. PQ BUS NUMBERS WHERE   | V |    IS TO BE FOUND  .
C           ENTER EACH BUS NUMBER ON A SEPARATE CARD USING
C           (F5.0) FORMAT .
C
C       INSERT A BLANK CARD
C
C       9. FOR EACH DESIRED REAL AND REACTIVE POWER TRANSFER
C          ENTER ON A SEPARATE CARD  , USING  (F10.5) FORMAT
C          RECEIVING BUS NUMBER , SENDING BUS NUMBER ,
C          1/2 CHARGING ADMITTANCE
C
C       INSERT A BLANK CARD
C
C
C
C
C
C
C                 MAIN  PROGRAM
C               -------------------
C
        COMMON N,N1,N21,N2,PV,PV1,PV2
        COMPLEX*16 Y(5,5),YSHT,ZSER,CONE
        REAL*8 UR(5,5),DR(5,5),DL(5,5),BETA(10),ZINJ(50,10)
        REAL*8 WR(108),XO(10),XK(10,10),LUTRSP(10,10),
       + R(10,10),C,XC,RES,XL,APXLIN,APXQDR,ZCZ,CZ,IPTRSP(10)
        INTEGER N,PV,PV1,PV2,BS,BB,SELECT(6),IPRINT(5)
C
        READ , E , F
        N = E
```

```
        PV = F
        N1 = N + 1
        N2 = 2 * N
        N21 = N2 - 1
        PV1 = PV + 1
        PV2 = PV + 2
        NWR = N21 * N21  +  3 * N21
C
C
C
        READ,(SELECT(I) , I=1,6)
        READ,(IPRINT(I) , I=1,5)
C
C       ** READ INJECTION VECTOR
C
           K = 1
1          I = 1
2          READ (5,3)  ZINJ(K,I)
3          FORMAT (F10.5)
           IF (ZINJ(K,I) .EQ. 0) GO TO 5
           IF (ZINJ(K,I) .GT. 100) GO TO 4
           I = I + 1
           GO TO 2
4          K = K + 1
           GO TO 1
5          CONTINUE
           INJ = K
C
C       ** READ BASE VOLTAGES
C
        DO 7 I=1,N2
        READ (5,6) XO(I)
6       FORMAT (F10.5)
7       CONTINUE
C          ** THIS IS USED TO READ A BLANK CARD
           READ (5,6) CBLANK
        WRITE (6,8)
8       FORMAT ('1',T2,' BASE VOLTAGES USED IN APPROXIMATION ',
       +         /T3,35('='),///)
        WRITE (6,9) (II,XO(II),II=1,N2)
9       FORMAT ('0',T1,4(4X,'V(',I3,') = ',F8.4,6X),/)
C
C       ** THIS LOOP FINDS THE ADMITTANCE MATRIX
C
C
            DO 10  I=1 , N
            DO 10  J=1 , N
10          Y(I,J) = DCMPLX(0.D0,0.D0)
            IF (IPRINT(2) .NE. 1) GO TO 12
            WRITE(6,26)
12          NLN = 0
```

```
14          READ (5,16) S,B,RES,XL,XC
16          FORMAT(5F10.5)
            IF (S .EQ. 0) GO TO 20
            NLN = NLN + 1
            YSHT= DCMPLX (0.D0,XC)
            ZSER= DCMPLX (RES,XL)
            IF(IPRINT(2) .NE. 1) GO TO 18
            WRITE(6,28) S,B,RES,XL,XC
18          M = S
            L = B
            ZSER = 1.D0 / ZSER
            Y(L,L) = Y(L,L) + ZSER + YSHT
            Y(M,M) = Y(M,M) + ZSER + YSHT
            Y(L,M) = Y(L,M) - ZSER
            Y(M,L) = Y(L,M)
            GO TO 14
20          IF(IPRINT(2) .NE.1) GO TO 22
            WRITE(6,30) NLN,N,PV
22          IF(IPRINT(3) .NE. 1) GO TO 35
            WRITE(6,32)
24          WRITE (6,34)  ((I,J,Y(I,J),J=1,N),I=1,N)
26          FORMAT('1','LINE DATA'/1X,9('=')///T8,
     +        'BUS NO. JOINS BUS NO. ',T36,'R  P.U.',T50,
     +        'XL  P.U.',T64,'YSHT  P.U.',/,T8,7('-'),T22,
     +        7('-'),T35,'---',T49,'----',T63,'-----'///)
28          FORMAT(8X,F4.0,10X,F4.0,1X,3F14.4/)
30          FORMAT(///////,T5,'THERE ARE ',I3,
     +              '   LINES IN THE SYSTEM',/T5,'THERE ARE ',
     +           I3,'  BUSES IN THE SYSTEM , AMONG WHICH ',
     +           I3,'  ARE  PV  BUSES ',//////)
32          FORMAT('1',T2,' BUS ADMITTANCE MATRIX ELEMENTS '
     +           /T3,30('='),///)
34          FORMAT(' ',T1,3(4X,'Y(',I3,',',I3,' ) = ',F7.3,
     +           ' +J ',F7.3,7X),/)
35      CONTINUE
C
C       ** THE FOLLOWING STEPS PRINT THE VOLTAGE PROFIL WHEN
C          DESIRED
            IF (SELECT(6) .EQ. 1) GO TO 51
            IF (IPRINT(1) .EQ. 0) GO TO 51
            WRITE (6,40)
40          FORMAT ('1',T2,'THE VOLTAGE PROFILS',/19('='),////)
            DO 50  MM=1,INJ
            WRITE (6,44)  (II,ZINJ(MM,II),II=1,N21)
44          FORMAT ('0',T1,4(4X,'V(',I3,') = ',F8.4,6X),/)
            WRITE (6,46)
46          FORMAT ('0',//////)
50          CONTINUE
C
C       **FIND 1/2 (JACOBIAN).DENOTED BY LUTRSP AT THIS STAGE.
C
```

```
51          CALL YAQUOB(Y,XO,UR,DR,DL,LUTRSP,WR,NWR)
C
C      **FIND THE TRANSPOSE OF THE JACOBIAN
C
                 DO 52 I=1,N2
                 DO 52 J=1,I
                 C = LUTRSP(I,J)
                 LUTRSP(I,J) = LUTRSP(J,I)
52               LUTRSP(J,I) = C
C
C      **ELEMINATE ROW AND COLUMN CORRESPONDING TO SLACK BUS
C        VOLTAGE .
C
53          CALL SHIFT (LUTRSP)
C
C      **LU DECOMPOSITION FOR MATRIX LUTRSP
C
            CALL LUDATF (LUTRSP,LUTRSP,N21,N2,6,CZ,ZCZ,IPTRSP,
     +                    IPTRSP,C,IER)
C
C
55     CONE = DCMPLX(0.D0,1.0D0)
       IF (SELECT(6) .EQ. 1) GO TO 71
C
C      ** IN CASE THE VOLTAGE PROFIL HAS BEEN SELECTED AS
C         INPUT DATA , THE FOLLOWING STEPS DETERMINE THE
C         INJECTION VECTOR USING THE FACT THAT  Z = L(X)   X .
C
              DO 70  I=1,INJ
              DO 57  J=1,N
              BETA(J) = ZINJ(I,J)
57            ZINJ(I,J) = 0.D0
              BETA(N1) = 0.D0
              DO 58  J=N1,N21
              K = J + 1
              BETA(K) = ZINJ(I,J)
58            ZINJ(I,J) = 0.D0
                 CALL YAQUOB(Y,BETA,UR,DR,DL,XK,WR,NWR)
                 CALL SHIFT(XK)
              DO 59  J=N1,N21
              K = J + 1
59            BETA(J)=BETA(K)
                 DO 62  MM=1,N21
                 DO 62  KK=1,N21
62               ZINJ(I,MM)= ZINJ(I,MM) + XK(MM,KK) * BETA(KK)
70            CONTINUE
C
C      ** TO PRINT INJECTION VECTORS
71     DO 75 MM=1,INJ
       WRITE (6,72) MM
72     FORMAT (/////,'0',T2,'THE INJECTION VECTOR NUMBER ',
```

```
     +          I3,/32('='),////)
      WRITE (6,74) (II,ZINJ(MM,II),II=1,N21)
74    FORMAT ('0',T1,4(4X,'Z(',I3,') = ',F8.4,6X),/)
75    CONTINUE
C
C     **SET  DR = REAL OF  Y  ADMITTANCE
C       SET  DL = IMAG OF  Y  ADMITTANCE
C
100         DO 120 K=1,N
            DO 120 I=1,N
            DR(I,K)=Y(I,K)
  120       DL(I,K)=-CONE*Y(I,K)
C
C     **REDUCE THE DIMENSION OF THE BASE VOLTAGE VECTOR FROM
C       2N  TO 2N-1
C
              DO 125 I=N1,N21
  125         XO(I)=XO(I+1)
C
C     **INITIALIZE   R   AND   XK
C
           C=  0.5D0
C
      IF (SELECT(1) .NE. 1) GO TO 199
C
           DO 130 I=1,N2
           DO 130 K=1,N2
           R (I,K)=0.D0
  130      XK(I,K)=0.D0
C
C     **THIS SUBROUTINE FINDS THE R MATRIX FOR P GENERATION
C       AT SLACK BUS.
C
            CALL PLACE ( R,DR,DL,C,1,1)
C
C     ** THIS STEP IS IDENTICAL TO THE PRECEEDING STEP   .
C        SETS XK = R .
C        . R WILL BE USED IN THE CALCULATION OF THE B VECTOR
C        . XK WILL BE USED IN THE CALCULATION OF THE C MATRIX
C
             CALL PLACE (XK,DR,DL,C,1,1)
C
C     ** THE  B  VECTOR AND THE  C  MATRIX  WILL BE FOUND FOR
C        THE P  GENERATION AT SLACK BUS .
C
             CALL SAVE (XO,R,WR,XK,DR,DL,NWR,BETA,0,SELECT,
     +                  LUTRSP,IPTRSP)
C
C     ** THE FOLLOWING STEPS CONCERN THE PRINTOUT
C
            IF (IPRINT(5) .NE. 1) GO TO 160
```

```
              WRITE (6,135)
135           FORMAT ('1',T2,'THE  B  VECTOR FOR SLACK BUS',
     +            ' REAL POWER GENERATION ',/T2,50('='),////)
              WRITE (6,140) (II , BETA(II),II=1,N21)
140           FORMAT ('0',T1,4(4X,'B(',I3,') = ',F8.4,6X))
              IF (SELECT(5) .EQ. 0) GO TO 160
              WRITE (6,145)
145           FORMAT ('1',T2,'THE  C  MATRIX FOR  SLACK BUS',
     +              'REAL ','POWER GENERATION',/T2,51('='),////)
              WRITE (6,150) ((II,JJ,XK(II,JJ),JJ=1,N21),II=1,N21)
150           FORMAT(' ',T1,4(4X,'C(',I3,',',I3,') = ',F8.4,
     +                7X),/)
160           IF (IPRINT(4) .NE. 1) GO TO 179
170           PUNCH 171
171           FORMAT ('BETA FOR REAL POWER GENERATION AT ',
     +                'SLACK BUS',I3)
              PUNCH 172 , (BETA(II) , II=1,N21)
172           FORMAT (6(F8.4,3X))
              IF (SELECT(5) .EQ. 0) GO TO 160
              PUNCH 173
173           FORMAT ('C  FOR  REAL POWER GENERATION AT ',
     +                'SLACK BUS ',I3)
              PUNCH 174 , ((XK(II,JJ),JJ=1,N21),II=1,N21)
174           FORMAT (6(F8.4,3X))
C
C
C           ** TO FIND THE LINEAR APPROXIMATION
179           DO 198  MM=1,INJ
              WRITE (6,72) MM
190           APXLIN = 0.D0
              APXQDR = 0.D0
              DO 191  KK=1,N21
191           APXLIN = APXLIN + BETA(KK) * ZINJ(MM,KK)
              WRITE (6,192) APXLIN
192           FORMAT ('0',T2,'*** THE LINEAR APPROXIMATION OF',
     +         ' THE ','REAL POWER GENERATION AT BUS  1   =  ',
     +         F8.4)
C             ** TO FIND THE SYSTEM'S TOTAL REAL POWER LOSS  ,
C             ADD THE REAL SLACK BUS POWER TO THE SPECIFIED REAL
C             INJECTIONS
              SUMINJ = 0.D0
              DO 182  JJ=N1,N21
182           SUMINJ = SUMINJ + ZINJ(MM,JJ)
              TOTLOS = SUMINJ + APXLIN
              WRITE (6,183) TOTLOS
183           FORMAT (//,'0',T2,'*** TOTAL LOSSES = ',F8.4)
              IF (SELECT(5) .NE. 1) GO TO 199
C           ** TO FIND THE QUADRATIC APPROXIMATION
              ZCZ =0.D0
              DO 194 II=1,N21
              CZ = 0.D0
```

```
          DO 193 JJ=1,N21
193       CZ = CZ + ZINJ(MM,JJ) * XK(II,JJ)
194       ZCZ = CZ * ZINJ(MM,II) + ZCZ
          APXQDR = ZCZ + APXLIN
          WRITE (6,195) APXQDR
195       FORMAT(///,'0',T2,'***  THE QUADRATIC APPROXIMATION'
     +       ' OF THE ','REAL POWER GENERATION AT BUS  1  = '
     +        ,F8.4)
C         ** TO FIND THE SYSTEM'S TOTAL REAL POWER LOSS ,
C            ADD THE REAL SLACK BUS POWER TO THE SPECIFIED
C            REAL POWER INJECTIONS .
             QDRLOS = SUMINJ + APXQDR
             WRITE (6,197) QDRLOS
197          FORMAT (//,'0',T2,'*** TOTAL LOSSES = ',F8.4)
198       CONTINUE
C
C
C
199            IF (SELECT(2) .NE. 1) GO TO 290
C
C     **THE PURPOSE OF THIS LOOP IS TO FIND THE  B  VECTOR
C       AND THE  C  MATRIX FOR THE  Q  GENERATION AT THE
C       SLACK BUS AND AT PV BUSES  .
C
C           INITIALIZE  R  ,  XK
C
C     ** READ DESIRED PV BUS
C
200   READ (5,201) PVBUS
201   FORMAT(F5.0)
      J = PVBUS
      IF (J .EQ. 0) GO TO 290
C
            DO 210 I=1,N2
            DO 210 K=1,N2
            R(I,K)=0.D0
210         XK(I,K)=0.D0
C
C      **   COMPUTE  R  MATRIX
C
            CALL PLACE ( R,DR,DL,C,J,2)
C
C      **   XK  AT THIS STEP WILL BE IDENTICAL TO  R  ABOVE
C
            CALL PLACE (XK,DR,DL,C,J,2)
C
C        ** COMPUTES  B  VECTOR AND  C  MATRIX  FOR THE  Q
C           INJECTION AT SLACK BUS , AND AT DESIRED  PV
C           BUSES .
C
            CALL SAVE (XO,R,WR,XK,DR,DL,NWR,BETA,0,SELECT,
```

```
     +                 LUTRSP,IPTRSP)
C
C
C     ** THE FOLLOWING STATEMENTS CONCERN THE PRINTOUT
C
              IF (IPRINT(5) .NE. 1) GO TO 250
              WRITE (6,220) J
220           FORMAT ('1',T2,'THE  B  VECTOR FOR  Q  ',
     +           'GENERATION AT BUS ',I3,/T2,44('='),////)
              WRITE (6,230)  (II , BETA(II),II=1,N21)
230           FORMAT ('0',T1,4(4X,'B(',I3,') = ',F8.4,6X))
              IF (SELECT(5) .EQ. 0) GO TO 250
              WRITE (6,235) J
235           FORMAT ('1',T2,'THE  C  MATRIX FOR  Q  ',
     +          'GENERATION AT BUS ',I3,/,T2,44('='),////)
              WRITE (6,240)  ((II,JJ,XK(II,JJ),JJ=1,N21),
     +                                 II=1,N21)
240           FORMAT('0',T1,4(4X,'C(',I3,',',I3,') = ',F8.4,
     +                 6X),/)
250           IF (IPRINT(4) .NE. 1) GO TO 275
270           PUNCH 271 , J
271           FORMAT ('BETA FOR  Q  GENERATION AT BUS ',I3)
              PUNCH 272 , (BETA(II) , II=1,N21)
272           FORMAT (6(F8.4,3X))
              IF(SELECT(5) .EQ. 0) GO TO 275
              PUNCH 273 , J
273           FORMAT ('C  FOR  Q  GENERATION AT BUS ',I3)
              PUNCH 274 , ((XK(II,JJ),JJ=1,N21),II=1,N21)
274           FORMAT (6(F8.4,3X))
C
C
C       ** TO FIND THE LINEAR APPROXIMATION
275       DO 288  MM=1,INJ
          WRITE (6,72) MM
280       APXLIN = 0.D0
          APXQDR = 0.D0
          DO 281 KK=1,N21
281       APXLIN = APXLIN + BETA(KK)  * ZINJ(MM,KK)
          WRITE (6,282) J , APXLIN
282       FORMAT ('0',T2,'*** THE LINEAR APPROXIMATION OF',
     +      ' THE  Q  GENERATION  AT BUS ',I3,' = ',F8.4,///)
          IF (SELECT(5) .NE. 1) GO TO 289
C       ** TO FIND THE QUADRATIC APPROXIMATION
          ZCZ =0.D0
          DO 284 II=1,N21
          CZ = 0.D0
          DO 283 JJ=1,N21
283       CZ = CZ + ZINJ(MM,JJ) * XK(II,JJ)
284       ZCZ = CZ * ZINJ(MM,II) + ZCZ
          APXQDR = ZCZ + APXLIN
          WRITE (6,285) J , APXQDR
```

```
285         FORMAT('0',T2,'*** THE QUADRATIC APPROXIMATION OF',
     +        ' THE  Q  GENERATION AT BUS ',I3,' = ',F8.4)
C
288         CONTINUE
C
289      GO TO 200
C
290      IF (SELECT(3) .NE. 1)GO TO 400
C
C        ** THIS LOOP FINDS THE B VECTOR , AND THE C MATRIX FOR
C           THE SQUARE OF THE VOLTAGE MAGNITUDE AT DESIRED  PQ
C           BUSES .
C
295      READ (5,296) PQBUS
296      FORMAT(F5.0)
         M = PQBUS
         IF (M .EQ. 0) GO TO 400
               DO 300 I=1,N2
               DO 300 K=1,N2
               R (I,K)=0.D0
  300          XK(I,K)=0.D0
               L=M+N
               XK(M,M)=1.0D0
               XK(L,L)=1.0D0
               R (M,M)=1.0D0
               R (L,L)=1.0D0
C
C
C
C          **  THIS SUBROUTINE FINDS THE  C  MATRIX , AND THE B
C              VECTOR
C
               CALL SAVE (XO,R,WR,XK,DR,DL,NWR,BETA,0,SELECT,
     +                    LUTRSP,IPTRSP)
C
C
C        THE FOLLOWING STEPS CONCERN THE OUTPUT
C
            IF (IPRINT(5) .NE. 1) GO TO 370
            WRITE (6,320) M
320         FORMAT ('1',T2,'THE  B  VECTOR FOR  THE VOLTAGE ',
     +      'SQUARED AT BUS ',I3,/T2,50('='),////)
            WRITE (6,330)  (II , BETA(II),II=1,N21)
330         FORMAT ('0',T1,4(4X,'B(',I3,') = ',F8.4,6X),/)
            IF (SELECT(5) .EQ. 0) GO TO 370
            WRITE (6,335) M
335         FORMAT ('1',T2,'THE  C  MATRIX FOR  THE VOLTAGE ',
     +              'SQUARED AT BUS ',I3,/,T2,53('='),////)
            WRITE (6,340) ((II,JJ,XK(II,JJ),JJ=1,N21),II=1,N21)
340         FORMAT('0',T1,4(4X,'C(',I3,',',I3,') = ',F8.4,6X),/)
370         IF (IPRINT(4) .NE. 1) GO TO 375
```

```
            PUNCH 371 , M
371         FORMAT ('BETA FOR   V   SQUARED AT BUS ',I3)
            PUNCH 372 , (BETA(II) , II=1,N21)
372         FORMAT ('0',6(F8.4,3X))
            IF(SELECT(5) .EQ. 0) GO TO 375
            PUNCH 373 , M
373         FORMAT ('C  FOR   V   SQUARED AT BUS ',I3)
            PUNCH 374 , ((XK(II,JJ),JJ=1,N21),II=1,N21)
374         FORMAT (6(F8.4,3X))
C
C         ** TO FIND THE LINEAR APPROXIMATION
375         DO 388  MM=1,INJ
            WRITE (6,72) MM
380         APXLIN = 0.D0
            APXQDR = 0.D0
            DO 381 KK=1,N21
381         APXLIN = APXLIN + BETA(KK) * ZINJ(MM,KK)
            WRITE (6,382) M , APXLIN
382         FORMAT ('0',T2,'*** THE LINEAR APPROXIMATION OF ',
     +        'THE VOLTAGE SQUARED AT BUS ',I3,' = ',F8.4,///)
            IF (SELECT(5) .NE. 1) GO TO 389
C         ** TO FIND THE QUADRATIC APPROXIMATION
            ZCZ = 0.D0
            DO 384 II=1,N21
            CZ = 0.D0
            DO 383 JJ=1,N21
383         CZ = CZ + ZINJ(MM,JJ) * XK(II,JJ)
384         ZCZ = CZ * ZINJ(MM,II) + ZCZ
            APXQDR = ZCZ + APXLIN
            WRITE (6,385) M , APXQDR
385         FORMAT('0',T2,'*** THE QUADRATIC APPROXIMATION OF',
     +        ' THE VOLTAGE SQUARED AT BUS ',I3,' = ',F8.4)
C
388       CONTINUE
C
389     GO TO 295
C
C
400     IF(SELECT(4) .NE. 1) GO TO 700
C
C
        CALL PWRTRF(BS,BB,YSHT,DR,DL,XO,BETA,XK,R,WR,NWR,NOPT,
     +       ZINJ,SELECT,IPRINT,INJ,LUTRSP,IPTRSP)
C
C
700     WRITE (6,701)
701     FORMAT ('1')
C
C
        STOP
        END
```

```
C
C
C
C
C
C
C
C     THIS SUBROUTINE FINDS THE B VECTOR AND THE C MATRIX
C     FOR REAL  AND/OR  REACTIVE POWER TRANSFER FROM BUS M
C     TO BUS  N .
C
C
C
C
      SUBROUTINE PWRTRF (BS,BB,YSHT,G,B,XO,BETA,XK,R,WR,NWR,
     +            NOPT,ZINJ,SELECT,IPRINT,INJ,LUTRSP,IPTRSP)
      COMMON N,N1,N21,N2,PV,PV1,PV2
      COMPLEX*16 YSHT
      REAL*8 XO(N2),G(N,N),B(N,N),BETA(N2),WR(NWR),XK(N2,N2),
     +     R(N2,N2)
      REAL*8 ZINJ(50,N2),A1,A2,A3,A4,C,C1,C2,C3,C4,SHT,
      REAL*8 APXLIN,APXQDR,ZCZ,CZ,LUTRSP(N2,N2),IPTRSP(N2)
      INTEGER BS,BB,PV,PV1,PV2,IPRINT(5),SELECT(6)
25    READ (5,10) RECEIV ,SEND ,SHT
10    FORMAT (3F10.5)
      IF (RECEIV .EQ. 0) GO TO 400
      BS = RECEIV
      BB = SEND
      IF (BS .EQ. 0) GO TO 400
      YSHT = DCMPLX(0.D0,SHT)
      YSHT = 2 * YSHT
      NS=N+BS
      NB=N+BB
      NS1=NS-1
      NB1=NB-1
      NOPT = 1
20    DO 1 I=1,N2
      DO 1 J=1,N2
      R (I,J)=0.D0
  1   XK(I,J)=0.D0
      IF (NOPT.NE.1) GO TO 2
C
C     **THIS STEP IS EXECUTED WHEN THE REAL POWER TRANSFER IS
C       DESIRED
C
          A1= 0.5D0 * G(BS,BB)
          A2= 0.5D0 * B(BS,BB)
          A3=-G(BS,BB)+ YSHT
          GO TO 3
C
C     **THIS STEP IS EXECUTED WHEN THE REACTIVE POWER
```

```
C       TRANSFER IS DESIRED.
C
  2        A1=-0.5D0 * B(BS,BB)
           A2= 0.5D0 * G(BS,BB)
           A3=B(BS,BB) + YSHT*DCMPLX(0.D0,0.5D0)
C
C     **THIS STEP FINDS THE  R  MATRIX            ( XK = R )
C
  3   XK(BS,BS)=A3
      R (BS,BS)=A3
      XK(NS,NS)=A3
      R (NS,NS)=A3
      XK(BS,BB)=A1
      R (BS,BB)=A1
      XK(BB,BS)=A1
      R (BB,BS)=A1
      XK(BS,NB)=-A2
      R (BS,NB)=-A2
      XK(NB,BS)=-A2
      R (NB,BS)=-A2
      XK(BB,NS)= A2
      R (BB,NS)= A2
      XK(NS,BB)= A2
      R (NS,BB)= A2
      XK(NS,NB)=A1
      R (NS,NB)=A1
      XK(NB,NS)=A1
      R (NB,NS)=A1
C
C
C     ** FIND THE  C  MATRIX  , AND THE  B  VECTOR
C
  7   CALL SAVE (XO,R,WR,XK,G,B,NWR,BETA,0,SELECT,LUTRSP,IPTRSP)
C
C     ** THE FOLLOWING STATEMENTS CONCERN THE OUTPUT
C
         IF (IPRINT(5) .NE. 1) GO TO 120
         IF (NOPT .EQ. 2) GO TO 40
         WRITE (6,30) BS,BB
30       FORMAT ('1',T2,'THE  B  VECTOR  FOR THE REAL ',
     +   'POWER','FLOW LEAVING BUS ',I3,'  TOWARDS BUS ',I3,
     +   /T2,72('='),////)
         GO TO 60
40       WRITE (6,50) BS,BB
50       FORMAT ('1',T2,'THE  B  VECTOR FOR REACTIVE POWER',
     +           ' FLOW LEAVING BUS ',I3,' TOWARDS BUS ',I3,
     +           /T2,70('='),////)
60       WRITE (6,70) (II , BETA(II),II=1,N21)
70       FORMAT ('0',T1,4(4X,'B(',I3,') = ',F8.4,6X),/)
         IF (SELECT(5) .EQ. 0)GO TO 120
         IF (NOPT .EQ. 2) GO TO 85
```

```
      WRITE (6,80) BS ,BB
80    FORMAT ('1',T2,'THE  C  MATRIX FOR  REAL POWER ',
     +         'FLOW','LEAVING BUS ',I3,' TOWARDS BUS ',
     +         I3,/,T2,67('='),////)
      GO TO 100
85    WRITE (6,90) BS ,BB
90    FORMAT ('1',T2,'THE  C  MATRIX FOR REACTIVE POWER',
     +         ' FLOW LEAVING BUS ',I3,' TOWARDS BUS ',
     +         I3,/T2,70('='),////)
100   WRITE (6,110) ((II,JJ,XK(II,JJ),JJ=1,N21),
     +                                  II=1,N21)
110   FORMAT('0',T1,4(4X,'C(',I3,',',I3,') = ',F8.4,
            6X)/)
120   IF (IPRINT(4) .NE. 1) GO TO 222
      IF (NOPT.EQ. 2) GO TO 140
      PUNCH 130 , BS ,BB
130   FORMAT ('BETA FOR  REAL POWER TRANSFER LEAVING',
     +        ' BUS  ',I3,' TOWARDS BUS ',I3)
      GO TO 160
140   PUNCH 150 , BS,BB
150   FORMAT ('BETA FOR REACTIVE POWER TRANSFER',
     +         'LEAVING BUS ', I3,' TOWARDS BUS ',I3)
160   PUNCH 170 , (BETA(II) , II=1,N21)
170   FORMAT (6(F8.4,3X))
      IF (SELECT(5) .EQ. 0) GO TO 222
      IF (NOPT .EQ. 2) GO TO 190
      PUNCH 180 , BS,BB
180   FORMAT ('C  FOR  REAL POWER TRANSFER LEAVING ',
     +         'BUS', I3,' TOWARDS BUS ',I3)
      GO TO 210
190   PUNCH 200, BS,BB
200   FORMAT (' C  MATRIX FOR REAL POWER TRANSFER ',
     +         'LEAVING BUS ',I3,' TOWARDS BUS ',I3)
210   PUNCH 220 , ((XK(II,JJ),JJ=1,N21),II=1,N21)
220   FORMAT (6(F8.4,3X))
C
C
C    ** TO FIND THE LINEAR APPROXIMATION
222   DO 250   MM=1,INJ
      WRITE(6,223) MM
223   FORMAT(/////,'0',T2,'THE INJECTION VECTOR NUMBER '
     +    ,I3,/32('='),///)
230   APXLIN = 0.D0
      APXQDR = 0.D0
      DO 231 KK=1,N21
231   APXLIN = APXLIN + BETA(KK)  * ZINJ(MM,KK)
      IF (NOPT .EQ. 2) GO TO 233
      WRITE (6,232) BS , BB , APXLIN
232   FORMAT ('0',T2,'THE LINEAR APPROXIMATION FOR THE',
     +         ' REAL POWER FLOW LEAVING BUS ',I3,
     +         ' TOWARDS BUS ',I3,' = ',F8.4,///)
```

```
            GO TO 235
233         WRITE (6,234) BS , BB , APXLIN
234         FORMAT ('0',T2,'THE LINEAR APPROXIMATION FOR THE',
     +            ' REACTIVE POWER FLOW LEAVING BUS ',I3,
     +            ' TOWARDS BUS ',I3,' = ',F8.4,///)
235         IF (SELECT(5) .NE. 1) GO TO 250
            ZCZ =0.D0
            DO 237 II=1,N21
            CZ = 0.D0
            DO 236 JJ=1,N21
236         CZ = CZ + ZINJ(MM,JJ) * XK(II,JJ)
237         ZCZ = CZ * ZINJ(MM,II) + ZCZ
            APXQDR = ZCZ + APXLIN
            IF (NOPT .EQ. 2) GO TO 240
            WRITE (6,238) BS , BB , APXQDR
238         FORMAT ('0',T2,'THE QUADRATIC APPROXIMATION FOR',
     +            ' THE REAL POWER FLOW LEAVING BUS ',I3,
     +             ' TOWARDS BUS ',I3,' = ,F8.4,///)
            GO TO 250
240         WRITE (6,241) BS , BB , APXQDR
241         FORMAT ('0',T2,'THE QUADRATIC APPROXIMATION FOR',
     +        'THE REACTIVE POWER FLOW LEAVING BUS ',I3,
     +        ' TOWARDS BUS ',I3,' = ',F8.4)
C
250         CONTINUE
C
290       IF (NOPT .EQ. 2) GO TO 300
      NOPT = 2
      GO TO 20
C
300     GO TO 25
C
400   RETURN
      END
C
C
C     THIS SUBROUTINE CALCULATES THE  B  VECTOR , AND THE
C     C MATRIX
C
C
C
      SUBROUTINE SAVE (XO,R,WR,XK,DR,DL,NWR,YO,NV,SELECT,LUTRSP,
     +                 IPTRSP)
      COMMON N,N1,N21,N2,PV,PV1,PV2
      REAL*8 XO(N2),R(N2,N2),XK(N2,N2),YO(N2),DR(N,N),DL(N,N)
      REAL*8 WR(NWR),C,LUTRSP(N2,N2),IPTRSP(N2)
      INTEGER PV,PV1,PV2,SELECT(6)
C
C     **ELIMINATE ROW AND COLUMN CORRESPONDING TO SLACK BUS
C       IMAGINARY VOLTAGE
C
```

```
      CALL SHIFT (R)
C
            DO 6 I=1,N21
            C=0.D0
            DO 5 J=1,N21
  5         C=C+XO(J)*R(J,I)
  6         WR(I)=C
C
C     **NOW   WR = X(0)   R
C
C     **USING LU DECOMPOSITION FIND   B = X   R    L(X ) = YO
C                                         .    0        0
C
        CALL LUELMF (LUTRSP,WR,IPTRSP,N21,N2,YO)
C
      IF (SELECT(5) .EQ. 0) GO TO 70
C
C     **NOW THAT  B  HAS BEEN FOUND , PROCEED TO FIND  C  .
C
C     **FIND  ( R - J(B) )
 25   DO 30 K=1,PV1
      J=K+N
      XK(K,K)=XK(K,K)-YO(K)
 30   XK(J,J)=XK(J,J)-YO(K)
      DO 40 K=2,N
      C=-0.5D0 * YO(K+N-1)
 40   CALL PLACE (XK,DR,DL,C,K,1)
      IF (PV2.GT.N) GO TO 51
C
           DO 50 K=PV2,N
           C=-0.5D0*YO(K)
 50        CALL PLACE (XK,DR,DL,C,K,2)
C
 51   CALL SHIFT (XK)
C     ** FINDS 1/4 (R)
         DO 54   LL=1,N21
         DO 54   LLL=1,N21
 54      XK(LL,LLL) =  0.25D0 * XK(LL,LLL)
C
C                               T  -1
C     ** FINDS    R =  ( L(X(0)   )        ( R  - J(B) )
C        THIS IS DONE USING LU DECOMPOSITION WHICH SOLVES
C        A SET OF LINEAR EQUATIONS  A X = B .
C                                                       T
C                               WHERE   A :(JACOBIAN)
C                                       B :R-J(B) =XK
C
C        THE IMSL ROUTINE MUST BE SOLVED WITH ONE COLUMN OF
C        MATRIX B AT A TIME , FOR ALL COLUMNS .
C
C           ..DEFINE COLUMN
```

```
56          DO 60 K=1,N21
            DO 58 I=1,N21
58          WR(I) = XK(I,K)
C
C         ..SOLVE
            CALL LUELMF (LUTRSP,WR,IPTRSP,N21,N2,WR)
C                                                 T
C         ..LET THE ANSWER BE A ROW OF MATRIX   R
            DO 60 I=1,N21
60          R(K,I) = WR(I)
C
C                            T  -1                 -1
C     ** FINDS   XK = C = (L(X )   ( R  - J(B) )   L(X )
C                              0                     0
C
C        I AM REQUIRED TO SOLVE AN EQUATION OF THE FORM
C        X  A  =  B .
C
C             T  T    T
C        ===> A  X  = B,      A :JACOBIAN
C                             B :R GIVEN BY ABOVE EQUATION
C                             X :THE SYMETRIC  C  MATRIX
C        SOLVE FOR ONE COLUMN OF X AT A TIME , USING LU
C        DECOMPOSITION
C                                  T
C             ..DEFINE COLUMN OF R
62              DO 68 K=1,N21
                DO 64 I=1,N21
64              WR(I) = R(I,K)
C
C             ..SOLVE
                  CALL LUELMF (LUTRSP,WR,IPTRSP,N21,N2,WR)
C
C             ..LET ANSWER BE A COLUMN OF MATRIX XK
                  DO 68 I=1,N21
68                XK(I,K) = WR(I)
C
C
70    RETURN
      END
C
C
C
      SUBROUTINE PLACE (XK,G,B,C,K,NOPT)
      COMMON N,N1,N21,N2,PV,PV1,PV2
      DOUBLE PRECISION XK(N2,N2),G(N,N),B(N,N),X1,X2,C
      INTEGER PV,PV1,PV2
      J=K+N
      DO 30 I=1,N
      L=I+N
      IF (NOPT.GT.1) GO TO 10
```

```
      X1=G(I,K)*C
      X2=B(I,K)*C
      GO TO 20
 10   X1=-B(I,K)*C
      X2= G(I,K)*C
 20   XK(I,K)=XK(I,K)+X1
      XK(L,K)=XK(L,K)-X2
      XK(I,J)=XK(I,J)+X2
      XK(L,J)=XK(L,J)+X1
      XK(K,I)=XK(K,I)+X1
      XK(K,L)=XK(K,L)-X2
      XK(J,I)=XK(J,I)+X2
 30   XK(J,L)=XK(J,L)+X1
      RETURN
      END
C
C
C
C     THIS SUBROUTINE REDUCES THE DIMENSION OF THE MATRIX
C     FROM  ( 2N* 2N )  TO    (2N-1)*(2N-1)
C
C
      SUBROUTINE SHIFT (A)
      COMMON N,N1,N21,N2,PV,PV1,PV2
      DOUBLE PRECISION A(N2,N2)
      DO 10 I=N1,N21
      DO 10 K= 1,N2
 10   A(I,K)=A(I+1,K)
      DO 20 K=N1,N21
      DO 20 I= 1,N2
 20   A(I,K)=A(I,K+1)
      RETURN
      END
C
C
C
C     THIS SUBROUTINE FINDS THE JACOBIAN OF THE SYSTEM
C
C
C
      SUBROUTINE YAQUOB(Y,XO,UR,DR,DL,XL,WR,NWR)
      COMMON N,N1,N21,N2,PV,PV1,PV2
      COMPLEX*16 Y(N,N),CONE,CX,CY,CZ,CS
      DOUBLE PRECISION XL(N2,N2),WR(NWR),XO(N2),UR(N,N)
     +                 DL(N,N),DR(N,N)
      REAL*8 A,B
      INTEGER PV,PV1,PV2
      CONE=DCMPLX(0.D0,1.0D0)
      DO 20 K=1,PV1
      DO 10 I=1,N
      XL(K,I)=0.D0
```

```
10   UR(K,I)=0.D0
     XL(K,K)=XO(K)
20   UR(K,K)=XO(K+N)
     DO 40 K=1,N
     J=K+N
     CS=DCMPLX(0.D0,0.D0)
     A=0.5D0*XO(K)
     B=-0.5D0*XO(J)
     CY=DCMPLX(A,B)
     DO 30 I=1,N
     A=XO(I)
     B=XO(I+N)
     CS=CS+Y(I,K)*DCMPLX(A,B)
     CX=CY*Y(I,K)
     DL(K,I)=CX
30   DR(K,I)=CONE*CX
     CS=0.5D0*CS
     WR(K)=CS
40   WR(J)=-CONE*CS
     IF (PV2.GT.N) GO TO 51
     DO 50 K=PV2,N
     DO 50 I=  1,N
     XL(K,I)=DR(K,I)
50   UR(K,I)=-DL(K,I)
51   DO 60 K=1,N
     J=K+N
     DL(K,K)=DL(K,K)+WR(K)
     DR(K,K)=DR(K,K)+WR(J)
     IF (K.LE.PV1) GO TO 60
     XL(K,K)=XL(K,K)-WR(J)
     UR(K,K)=UR(K,K)+WR(K)
60   CONTINUE
     DO 80 K=1,N
     J=K+N
     DO 70 I=1,N
     L=I+N
     XL(K,L)=UR(K,I)
     XL(J,I)=DL(K,I)
70   XL(J,L)=DR(K,I)
80   CONTINUE
     RETURN
     END
```

```
$DATA
5    2
1    1    1    1    1    1
0    1    1    0    1
1.1236
1.1025
1.0816
-0.050
-0.100
-0.492
-0.077
-0.04
-0.600
999
1.1236
1.1025
1.0816
-0.0490
-0.0980
0.4820
0.0750
-0.3920
-0.5880

1.06
1.04990
1.03948
1.03599
1.02214
0.000000
-0.01484
-0.03303
-0.04298
-0.06813

1.         2.         0.02       0.06       0.03
1.         3.         0.08       0.24       0.025
2.         3.         0.06       0.18       0.02
2.         4.         0.06       0.18       0.02
2.         5.         0.04       0.12       0.015
3.         4.         0.01       0.03       0.01
4.         5.         0.08       0.24       0.025

2.         4.         0.02
2.         5.         0.015
3.         4.         0.01

$
```

BASE VOLTAGES USED IN APPROXIMATION

V(1)= 1.0600	V(2)= 1.0499	V(3)= 1.0395	V(4)= 1.0360
V(5)= 1.0221	V(6)= 0.0	V(7)=-0.0148	V(8)=-0.0330
V(9)=-0.0430	V(10)=-0.0681		

LINE DATA

BUS NO. JOINS	BUS NO.	R P.U	XL P.U.	YSHT P.U.
1.	2.	0.0200	0.0600	0.0300
1.	3.	0.0800	0.2400	0.0250
2.	3.	0.0600	0.1800	0.0200
2.	4.	0.0600	0.1800	0.0200
2.	5.	0.0400	0.1200	0.0150
3.	4.	0.0100	0.0300	0.0100
4.	5.	0.0800	0.2400	0.0250

THERE ARE 7 LINES IN THE SYSTEM
THERE ARE 5 BUSES IN THE SYSTEM , 2 OF WHICH ARE PV

THE INJECTION VECTOR NUMBER 1
================================

Z(1)= 1.1236	Z(2)= 1.1025	Z(3)= 1.0816	Z(4)=-0.0500
Z(5)=-0.1000	Z(6)= 0.4920	Z(7)= 0.0770	Z(8)=-0.4000
Z(9)=-0.6000			

THE INJECTION VECTOR NUMBER 2
================================

Z(1)= 1.1236	Z(2)= 1.1025	Z(3)= 1.0816	Z(4)=-0.0490
Z(5)=-0.0980	Z(6)= 0.4820	Z(7)= 0.0750	Z(8)=-0.3920
Z(9)=-0.5880			

THE B VECTOR FOR THE REAL POWER FLOW
LEAVING BUS 2 TOWARDS BUS 4

B(1)=-0.0084	B(2)= 0.0181	B(3)=-0.0115	B(4)= 0.0021
B(5)=-0.0018	B(6)= 0.0569	B(7)=-0.2314	B(8)=-0.3101
B(9)=-0.0665			

THE C MATRIX FOR REAL POWER FLOW
LEAVING BUS 2 TOWARDS BUS 4

C(1,1)= 0.0030	C(1,2)= 0.0380	C(1,3)=-0.0371	C(1,4)=-0.0001
C(1,5)= 0.0001	C(1,6)=-0.0030	C(1,7)= 0.0127	C(1,8)= 0.0096
C(1,9)= 0.0012	C(2,1)= 0.0380	C(2,2)= 0.2081	C(2,3)=-0.2540
C(2,4)=-0.0049	C(2,5)= 0.0041	C(2,6)=-0.0090	C(2,7)=-0.0092
C(2,8)=-0.0230	C(2,9)= 0.0022	C(3,1)=-0.0371	C(3,2)=-0.2540
C(3,3)= 0.2969	C(3,4)= 0.0041	C(3,5)=-0.0032	C(3,6)= 0.0122
C(3,7)=-0.0022	C(3,8)= 0.0159	C(3,9)=-0.0019	C(4,1)=-0.0001
C(4,2)=-0.0049	C(4,3)= 0.0041	C(4,4)= 0.0022	C(4,5)= 0.0004
C(4,6)= 0.0004	C(4,7)=-0.0016	C(4,8)=-0.0012	C(4,9)=-0.0010
C(5,1)= 0.0001	C(5,2)= 0.0041	C(5,3)=-0.0032	C(5,4)= 0.0004
C(5,5)= 0.0019	C(5,6)=-0.0003	C(5,7)= 0.0012	C(5,8)= 0.0019
C(5,9)= 0.0003	C(6,1)=-0.0030	C(6,2)=-0.0090	C(6,3)= 0.0122
C(6,4)= 0.0004	C(6,5)=-0.0003	C(6,6)= 0.0000	C(6,7)=-0.0004
C(6,8)=-0.0003	C(6,9)=-0.0001	C(7,1)= 0.0127	C(7,2)=-0.0092
C(7,3)=-0.0022	C(7,4)=-0.0016	C(7,5)= 0.0012	C(7,6)=-0.0004
C(7,7)= 0.0041	C(7,8)= 0.0032	C(7,9)= 0.0009	C(8,1)= 0.0096
C(8,2)=-0.0230	C(8,3)= 0.0159	C(8,4)=-0.0012	C(8,5)= 0.0019
C(8,6)=-0.0003	C(8,7)= 0.0032	C(8,8)= 0.0047	C(8,9)= 0.0012
C(9,1)= 0.0012	C(9,2)= 0.0022	C(9,3)=-0.0019	C(9,4)=-0.0010
C(9,5)= 0.0003	C(9,6)=-0.0001	C(9,7)= 0.0009	C(9,8)= 0.0012
C(9,9)= 0.0021			

THE INJECTION VECTOR NUMBER 1

THE LINEAR APPROXIMATION FOR THE REAL POWER FLOW
LEAVING BUS 2 TOWARDS BUS 4 = 0.1724

THE QUADRATIC APPROXIMATION FOR THE REAL POWER FLOW
LEAVING BUS 2 TOWARDS BUS 4 = 0.1724

THE INJECTION VECTOR NUMBER 2
=================================

THE LINEAR APPROXIMATION FOR THE REAL POWER FLOW
LEAVING BUS 2 TOWARDS BUS 4 = 0.1690

THE QUADRATIC APPROXIMATION FOR THE REAL POWER FLOW
LEAVING BUS 2 TOWARDS BUS 4 = 0.1690

THE B VECTOR FOR THE REACTIVE POWER FLOW
LEAVING BUS 2 TOWARDS BUS 4

B(1)= 0.0025 B(2)= 2.1997 B(3)=-2.2349 B(4)=-0.1349

B(5)=-0.0450 B(6)=-0.0172 B(7)= 0.0700 B(8)= 0.0476

B(9)= 0.0009

THE C MATRIX FOR REACTIVE POWER FLOW
LEAVING BUS 2 TOWARDS BUS 4

C(1,1)=-0.0009	C(1,2)=-0.0107	C(1,3)= 0.0106	C(1,4)=-0.0000
C(1,5)=-0.0000	C(1,6)= 0.0009	C(1,7)=-0.0036	C(1,8)=-0.0026
C(1,9)=-0.0003	C(2,1)=-0.0107	C(2,2)= 0.4945	C(2,3)=-0.4848
C(2,4)=-0.0255	C(2,5)=-0.0036	C(2,6)=-0.0024	C(2,7)= 0.0236
C(2,8)= 0.0236	C(2,9)= 0.0028	C(3,1)= 0.0106	C(3,2)=-0.4848
C(3,3)= 0.4795	C(3,4)= 0.0263	C(3,5)= 0.0041	C(3,6)= 0.0007
C(3,7)=-0.0172	C(3,8)=-0.0166	C(3,9)= 0.0001	C(4,1)=-0.0000
C(4,2)=-0.0255	C(4,3)= 0.0263	C(4,4)= 0.0027	C(4,5)= 0.0006
C(4,6)= 0.0001	C(4,7)=-0.0004	C(4,8)=-0.0003	C(4,9)= 0.0003
C(5,1)=-0.0000	C(5,2)=-0.0036	C(5,3)= 0.0041	C(5,4)= 0.0006
C(5,5)= 0.0035	C(5,6)= 0.0001	C(5,7)=-0.0004	C(5,8)=-0.0005
C(5,9)= 0.0003	C(6,1)= 0.0009	C(6,2)=-0.0024	C(6,3)= 0.0007
C(6,4)= 0.0001	C(6,5)= 0.0001	C(6,6)= 0.0004	C(6,7)=-0.0015
C(6,8)=-0.0018	C(6,9)=-0.0002	C(7,1)=-0.0036	C(7,2)= 0.0236
C(7,3)=-0.0172	C(7,4)=-0.0004	C(7,5)=-0.0004	C(7,6)=-0.0015
C(7,7)= 0.0054	C(7,8)= 0.0069	C(7,9)= 0.0006	C(8,1)=-0.0026
C(8,2)= 0.0236	C(8,3)=-0.0166	C(8,4)=-0.0003	C(8,5)=-0.0005
C(8,6)=-0.0018	C(8,7)= 0.0069	C(8,8)= 0.0100	C(8,9)= 0.0013
C(9,1)=-0.0003	C(9,2)= 0.0028	C(9,3)= 0.0001	C(9,4)= 0.0003
C(9,5)= 0.0003	C(9,6)=-0.0002	C(9,7)= 0.0006	C(9,8)= 0.0013
C(9,9)= 0.0037			

THE INJECTION VECTOR NUMBER 1

THE LINEAR APPROXIMATION FOR THE REACTIVE POWER FLOW
LEAVING BUS 2 TOWARDS BUS 4 = -0.0007

THE QUADRATIC APPROXIMATION FOR THE REACTIVE POWER FLOW
LEAVING BUS 2 TOWARDS BUS 4 = -0.0007

THE INJECTION VECTOR NUMBER 2

THE LINEAR APPROXIMATION FOR THE REACTIVE POWER FLOW
LEAVING BUS 2 TOWARDS BUS 4 = -0.0005

THE QUADRATIC APPROXIMATION FOR THE REACTIVE POWER FLOW
LEAVING BUS 2 TOWARDS BUS 4 = -0.0005

THE B VECTOR FOR THE REAL POWER FLOW
LEAVING BUS 2 TOWARDS BUS 5

B(1)=-0.0042	B(2)= 0.0559	B(3)=-0.0596	B(4)=-0.0022
B(5)= 0.0012	B(6)= 0.0283	B(7)=-0.1150	B(8)=-0.1552
B(9)=-0.7232			

THE C MATRIX FOR REAL POWER FLOW
LEAVING BUS 2 TOWARDS BUS 5

C(1,1)= 0.0015	C(1,2)= 0.0189	C(1,3)=-0.0185	C(1,4)=-0.0000
C(1,5)=-0.0001	C(1,6)=-0.0015	C(1,7)= 0.0063	C(1,8)= 0.0048
C(1,9)= 0.0006	C(2,1)= 0.0189	C(2,2)= 0.0770	C(2,3)=-0.1200
C(2,4)=-0.0019	C(2,5)=-0.0115	C(2,6)=-0.0044	C(2,7)=-0.0048
C(2,8)=-0.0116	C(2,9)=-0.0340	C(3,1)=-0.0185	C(3,2)=-0.1200
C(3,3)= 0.1683	C(3,4)= 0.0031	C(3,5)= 0.0124	C(3,6)= 0.0061
C(3,7)=-0.0015	C(3,8)= 0.0078	C(3,9)= 0.0455	C(4,1)=-0.0000
C(4,2)=-0.0019	C(4,3)= 0.0031	C(4,4)= 0.0012	C(4,5)= 0.0011
C(4,6)= 0.0002	C(4,7)=-0.0008	C(4,8)=-0.0006	C(4,9)= 0.0023
C(5,1)=-0.0001	C(5,2)=-0.0115	C(5,3)= 0.0124	C(5,4)= 0.0011
C(5,5)= 0.0188	C(5,6)= 0.0010	C(5,7)=-0.0041	C(5,8)=-0.0051
C(5,9)= 0.0016	C(6,1)=-0.0015	C(6,2)=-0.0044	C(6,3)= 0.0061
C(6,4)= 0.0002	C(6,5)= 0.0010	C(6,6)= 0.0000	C(6,7)=-0.0002
C(6,8)=-0.0002	C(6,9)= 0.0000	C(7,1)= 0.0063	C(7,2)=-0.0048
C(7,3)=-0.0015	C(7,4)=-0.0008	C(7,5)=-0.0041	C(7,6)=-0.0002
C(7,7)= 0.0021	C(7,8)= 0.0017	C(7,9)= 0.0001	C(8,1)= 0.0048
C(8,2)=-0.0116	C(8,3)= 0.0078	C(8,4)=-0.0006	C(8,5)=-0.0051
C(8,6)=-0.0002	C(8,7)= 0.0017	C(8,8)= 0.0025	C(8,9)= 0.0011
C(9,1)= 0.0006	C(9,2)=-0.0340	C(9,3)= 0.0455	C(9,4)= 0.0023
C(9,5)= 0.0016	C(9,6)= 0.0000	C(9,7)= 0.0001	C(9,8)= 0.0011
C(9,9)= 0.0195			

THE INJECTION VECTOR NUMBER 1

THE LINEAR APPROXIMATION FOR THE REAL POWER FLOW
LEAVING BUS 2 TOWARDS BUS 5 = 0.4935

THE QUADRATIC APPROXIMATION FOR THE REAL POWER FLOW
LEAVING BUS 2 TOWARDS BUS 5 = 0.4935

THE INJECTION VECTOR NUMBER 2

THE LINEAR APPROXIMATION FOR THE REAL POWER FLOW
LEAVING BUS 2 TOWARDS BUS 5 = 0.4836

THE QUADRATIC APPROXIMATION FOR THE REAL POWER FLOW
LEAVING BUS 2 TOWARDS BUS 5 = 0.4836

THE B VECTOR FOR THE REACTIVE POWER FLOW
LEAVING BUS 2 TOWARDS BUS 5

B(1)= 0.0011	B(2)= 1.0500	B(3)=-1.1212	B(4)=-0.0676
B(5)=-0.7053	B(6)=-0.0072	B(7)= 0.0293	B(8)= 0.0161
B(9)=-0.0715			

THE C MATRIX FOR REACTIVE POWER FLOW
LEAVING BUS 2 TOWARDS BUS 5

C(1,1)=-0.0004	C(1,2)=-0.0044	C(1,3)= 0.0044	C(1,4)= -0.0000
C(1,5)=-0.0000	C(1,6)= 0.0004	C(1,7)=-0.0015	C(1,8)=-0.0011
C(1,9)= 0.0000	C(2,1)=-0.0044	C(2,2)= 0.2651	C(2,3)=-0.2443
C(2,4)=-0.0126	C(2,5)=-0.0363	C(2,6)=-0.0019	C(2,7)= 0.0135
C(2,8)= 0.0139	C(2,9)= 0.0365	C(3,1)= 0.0044	C(3,2)=-0.2443
C(3,3)= 0.2468	C(3,4)= 0.0132	C(3,5)= 0.0460	C(3,6)= 0.0005
C(3,7)=-0.0079	C(3,8)=-0.0069	C(3,9)=-0.0007	C(4,1)=-0.0000
C(4,2)=-0.0126	C(4,3)= 0.0132	C(4,4)= 0.0014	C(4,5)= 0.0030
C(4,6)= 0.0001	C(4,7)=-0.0002	C(4,8)=-0.0001	C(4,9)= 0.0000
C(5,1)=-0.0000	C(5,2)=-0.0363	C(5,3)= 0.0460	C(5,4)= 0.0030
C(5,5)= 0.0571	C(5,6)= 0.0000	C(5,7)=-0.0001	C(5,8)= 0.0008
C(5,9)= 0.0058	C(6,1)= 0.0004	C(6,2)=-0.0019	C(6,3)= 0.0005
C(6,4)= 0.0001	C(6,5)= 0.0000	C(6,6)= 0.0002	C(6,7)=-0.0008
C(6,8)=-0.0009	C(6,9)=-0.0013	C(7,1)=-0.0015	C(7,2)= 0.0135
C(7,3)=-0.0079	C(7,4)=-0.0002	C(7,5)=-0.0001	C(7,6)=-0.0008
C(7,7)= 0.0028	C(7,8)= 0.0035	C(7,9)= 0.0052	C(8,1)=-0.0011
C(8,2)= 0.0139	C(8,3)=-0.0069	C(8,4)=-0.0001	C(8,5)= 0.0008
C(8,6)=-0.0009	C(8,7)= 0.0035	C(8,8)= 0.0051	C(8,9)= 0.0072
C(9,1)= 0.0000	C(9,2)= 0.0365	C(9,3)=-0.0007	C(9,4)= 0.0000
C(9,5)= 0.0058	C(9,6)=-0.0013	C(9,7)= 0.0052	C(9,8)= 0.0072
C(9,9)= 0.0599			

THE INJECTION VECTOR NUMBER 1

THE LINEAR APPROXIMATION FOR THE REACTIVE POWER FLOW
LEAVING BUS 2 TOWARDS BUS 5 = 0.0552

THE QUADRATIC APPROXIMATION FOR THE REACTIVE POWER FLOW
LEAVING BUS 2 TOWARDS BUS 5 = 0.0552

THE INJECTION VECTOR NUMBER 2

THE LINEAR APPROXIMATION FOR THE REACTIVE POWER FLOW
LEAVING BUS 2 TOWARDS BUS 5 = 0.0530

THE QUADRATIC APPROXIMATION FOR THE REACTIVE POWER FLOW
LEAVING BUS 2 TOWARDS BUS 5 = 0.0530

THE B VECTOR FOR THE REAL POWER FLOW
LEAVING BUS 3 TOWARDS BUS 4

B(1)= 0.0124	B(2)=-0.0747	B(3)= 0.0596	B(4)=-0.0004
B(5)=-0.0040	B(6)=-0.0842	B(7)= 0.3423	B(8)=-0.5468
B(9)=-0.2447			

THE C MATRIX FOR REAL POWER FLOW
LEAVING BUS 3 TOWARDS BUS 4

C(1,1)=-0.0044	C(1,2)=-0.0555	C(1,3)= 0.0543	C(1,4)= 0.0001
C(1,5)=-0.0000	C(1,6)= 0.0044	C(1,7)=-0.0186	C(1,8)=-0.0141
C(1,9)=-0.0018	C(2,1)=-0.0555	C(2,2)= 0.2282	C(2,3)=-0.1361
C(2,4)= 0.0071	C(2,5)= 0.0107	C(2,6)= 0.0087	C(2,7)= 0.0325
C(2,8)= 0.0499	C(2,9)= 0.0456	C(3,1)= 0.0543	C(3,2)=-0.1361
C(3,3)= 0.0536	C(3,4)=-0.0067	C(3,5)=-0.0083	C(3,6)=-0.0139
C(3,7)=-0.0133	C(3,8)=-0.0337	C(3,9)=-0.0405	C(4,1)= 0.0001
C(4,2)= 0.0071	C(4,3)=-0.0067	C(4,4)= 0.0041	C(4,5)= 0.0010
C(4,6)=-0.0006	C(4,7)= 0.0024	C(4,8)= 0.0019	C(4,9)=-0.0008
C(5,1)=-0.0000	C(5,2)= 0.0107	C(5,3)=-0.0083	C(5,4)= 0.0010
C(5,5)= 0.0068	C(5,6)=-0.0007	C(5,7)= 0.0030	C(5,8)= 0.0035
C(5,9)= 0.0010	C(6,1)= 0.0044	C(6,2)= 0.0087	C(6,3)=-0.0139
C(6,4)=-0.0006	C(6,5)=-0.0007	C(6,6)= 0.0004	C(6,7)=-0.0008
C(6,8)=-0.0006	C(6,9)=-0.0000	C(7,1)=-0.0186	C(7,2)= 0.0325
C(7,3)=-0.0133	C(7,4)= 0.0024	C(7,5)= 0.0030	C(7,6)=-0.0008
C(7,7)=-0.0002	C(7,8)=-0.0002	C(7,9)=-0.0004	C(8,1)=-0.0141
C(8,2)= 0.0499	C(8,3)=-0.0337	C(8,4)= 0.0019	C(8,5)= 0.0035
C(8,6)=-0.0006	C(8,7)=-0.0002	C(8,8)= 0.0039	C(8,9)= 0.0008
C(9,1)=-0.0018	C(9,2)= 0.0456	C(9,3)=-0.0405	C(9,4)=-0.0008
C(9,5)= 0.0010	C(9,6)=-0.0000	C(9,7)=-0.0004	C(9,8)= 0.0008
C(9,9)= 0.0068			

THE INJECTION VECTOR NUMBER 1

THE LINEAR APPROXIMATION FOR THE REAL POWER FLOW
LEAVING BUS 3 TOWARDS BUS 4 = 0.3469

THE QUADRATIC APPROXIMATION FOR THE REAL POWER FLOW
LEAVING BUS 3 TOWARDS BUS 4 = 0.3469

THE INJECTION VECTOR NUMBER 2

THE LINEAR APPROXIMATION FOR THE REAL POWER FLOW
LEAVING BUS 3 TOWARDS BUS 4 = 0.3397

THE QUADRATIC APPROXIMATION FOR THE REAL POWER FLOW
LEAVING BUS 3 TOWARDS BUS 4 = 0.3397

THE B VECTOR FOR THE REACTIVE POWER FLOW
LEAVING BUS 3 TOWARDS BUS 4

B(1)=-0.0040	B(2)=-3.3275	B(3)= 3.2566	B(4)=-0.8022
B(5)=-0.2712	B(6)= 0.0274	B(7)=-0.1115	B(8)=-0.1012
B(9)=-0.0360			

THE C MATRIX FOR REACTIVE POWER FLOW
LEAVING BUS 3 TOWARDS BUS 4

C(1,1)= 0.0015	C(1,2)= 0.0191	C(1,3)=-0.0188	C(1,4)= 0.0000
C(1,5)= 0.0001	C(1,6)=-0.0015	C(1,7)= 0.0064	C(1,8)= 0.0047
C(1,9)= 0.0004	C(2,1)= 0.0191	C(2,2)= 0.7846	C(2,3)=-0.8056
C(2,4)= 0.0390	C(2,5)= 0.0500	C(2,6)=-0.0099	C(2,7)= 0.0183
C(2,8)= 0.0087	C(2,9)= 0.0021	C(3,1)=-0.0188	C(3,2)=-0.8056
C(3,3)= 0.8342	C(3,4)=-0.0377	C(3,5)=-0.0473	C(3,6)= 0.0120
C(3,7)=-0.0259	C(3,8)=-0.0064	C(3,9)= 0.0101	C(4,1)= 0.0000
C(4,2)= 0.0390	C(4,3)=-0.0377	C(4,4)= 0.0186	C(4,5)= 0.0042
C(4,6)=-0.0001	C(4,7)= 0.0006	C(4,8)= 0.0008	C(4,9)= 0.0010
C(5,1)= 0.0001	C(5,2)= 0.0500	C(5,3)=-0.0473	C(5,4)= 0.0042
C(5,5)= 0.0222	C(5,6)=-0.0002	C(5,7)= 0.0008	C(5,8)= 0.0005
C(5,9)= 0.0023	C(6,1)=-0.0015	C(6,2)=-0.0099	C(6,3)= 0.0120
C(6,4)=-0.0001	C(6,5)=-0.0002	C(6,6)= 0.0005	C(6,7)=-0.0022
C(6,8)=-0.0006	C(6,9)= 0.0011	C(7,1)= 0.0064	C(7,2)= 0.0183
C(7,3)=-0.0259	C(7,4)= 0.0006	C(7,5)= 0.0008	C(7,6)=-0.0022
C(7,7)= 0.0101	C(7,8)= 0.0035	C(7,9)=-0.0041	C(8,1)= 0.0047
C(8,2)= 0.0087	C(8,3)=-0.0064	C(8,4)= 0.0008	C(8,5)= 0.0005
C(8,6)=-0.0006	C(8,7)= 0.0035	C(8,8)= 0.0182	C(8,9)= 0.0008
C(9,1)= 0.0004	C(9,2)= 0.0021	C(9,3)= 0.0101	C(9,4)= 0.0010
C(9,5)= 0.0023	C(9,6)= 0.0011	C(9,7)=-0.0041	C(9,8)= 0.0008
C(9,9)= 0.0221			

THE INJECTION VECTOR NUMBER 1

THE LINEAR APPROXIMATION FOR THE REACTIVE POWER FLOW
LEAVING BUS 3 TOWARDS BUS 4 = -0.0166

THE QUADRATIC APPROXIMATION FOR THE REACTIVE POWER FLOW
LEAVING BUS 3 TOWARDS BUS 4 = -0.0166

THE INJECTION VECTOR NUMBER 2

THE LINEAR APPROXIMATION FOR THE REACTIVE POWER FLOW
LEAVING BUS 3 TOWARDS BUS 4 = -0.0192

THE QUADRATIC APPROXIMATION FOR THE REACTIVE POWER FLOW
LEAVING BUS 3 TOWARDS BUS 4 = -0.0192

APPENDIX C

LOAD FLOW PROGRAM

```
$WATFIV
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CC                                                          CC
CC                                                          CC
CC                                                          CC
CC                                                          CC
CC                                                          CC
CC     LOAD FLOW PROGRAM USING THE NEWTON RAPHSON METHOD    CC
CC                                                          CC
CC     THIS PROGRAM WAS WRITTEN BY  S.L. LOW [24]           CC
CC                                                          CC
CC                                                          CC
CC                                                          CC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C      THIS PROGRAM IS DESIGNED TO ACCOMODATE UP TO
C
C            50 BUSES
C      AND  100 LINES
C
C      LARGER NETWORKS MAY BE SIMULATED BY INCREASING THE
C      DIMENSIONS.
C
C
C      THE LINE DATA IS ENTERED FIRST, IN F10.5 FORMAT IN
C      THE SEQUENCE NODE NUMBER, NODE NUMBER, LINE
C      RESISTANCE,LINE REACTANCE AND ONE HALF LINE CHARGING
C      ADMITTANCE (ALL IN PER UNIT)
C
C      THE LAST CARD OF LINE DATA IS SEPARATED FROM THE
C      FOLLOWING DECK OF BUS DATA CARDS BY A BLANK CARD
C
C      NEXT, THE BUS DATA IS ENTERED IN F5.2 FORMAT IN THE
C      SEQUENCE   BUS NUMBER, BUS TYPE, VOLTAGE MAGNITUDE,
C      REAL POWER GENERATION,REACTIVE POWER GENERATION,
C      REAL POWER DEMAND, REACTIVE DEMAND,
C      INITIAL ESTIMATE OF BUS VOLTAGE MAGNITUDE AND ANGLE
C
C      THE LAST CARD OF BUS DATA MUST BE FOLLOWED BY A BLANK
```

```
C     CARD
C
C     THE BUSES MUST BE NUMBERED CONSECUTIVELY STARTING
C     FROM 1 HOWEVER THERE IS NO RESTRICTION ON GROUPING OF
C     THE TYPES OF BUSES E.G. THE SLACK BUS CAN BE NUMBERED
C     THE FIRST BUS, THE LAST BUS OR ANY NUMBER IN BETWEEN,
C     THE SAME APPLIES TO PQ AND PV BUSES
C
C     THE BUS TYPES ARE CODED AS FOLLOWS;
C        1.0    PQ BUS
C        2.0    PV BUS
C        3.0    SLACK BUS WHICH IS ALSO THE REFERENCE BUS
C
C
C
C
C
C     THE FOLLOWING ARE SOME PARAMETERS THAT CONTROL THE
C     EXECUTION OF THE PROGRAM       THESE PARAMETERS
C     SHOULD BE READ IN THE FOLLOWING SEQUENCE
C     IN 5F10.5 FORMAT BEFORE OTHER DATA IS READ
C
C
C     IREAD  --- INPUT DEVICE NUMBER FOR LINE AND BUS DATA
C     IWRITE --- OUTPUT DEVICE NUMBER FOR RESULTS
C     CRITER --- ACCURACY TO WHICH THE MAXIMUM MISMATCH MUST
C                SATISFY
C     JUPDAT --- HOW FREQUENTLY THE JACOBIAN MATRIX IS
C                UPDATED
C           0  INDICATES THE INITIAL JACOBIAN MATRIX IS
C              USED THROUGHOUT
C           1  INDICATES THE JACOBIAN MATRIX IS UPDATED
C              EVERY ITERATION
C           2  INIDCATES JACOBIAN MATRIX UPDATED EVERY TWO
C              ITERATIONS
C           N  INDICATES JACOBIAN MATRIX UPDATED EVERY  N
C              ITERATIONS
C     LOOP   --- MAXIMUM NUMBER OF ITERATIONS ALLOWED
C
C
      IMPLICIT REAL*8 (A-H,O-Z), INTEGER*2 (I-N)
      REAL*4 PARM(5)
      INTEGER*4 IREAD,IWRITE
      LOGICAL*1 OK
      COMMON /LOADFL/ VREAL(120),VIMAG(120),VMAGSQ(120),
     +CREAL(120),
     +CIMAG(120),CMAGLN(200),REALG(120),REACTG(120),
     +REALD(120),REACTD(120),MODBUS(120),NREF,NOGEN
      COMMON /NETWOK/ DIAYMR(120),DIAYMI(120),DATAYR(200),
     +DATAYI(200),
     +DATALN(200),LKSTYM(120),JCOLYM(400),LINKYM(400),
```

```
      +KONECT(120),
      +LORDER(120),NORDER(120),LINE,NFAULT
       COMMON /LUSOLV/ DELTAX(240),ERRORZ(240),DELTAG(120),
      +DELTAQ(120),
      +DIAGUT(240),DATAUT(500),DATALT(500),LKSTUT(240),
      +JROWUT(500),
      +LINKUT(500),IRSTUT(240),JCOLUT(500),IRSTLT(240),
      +JCOLLT(500),
      +JCOLJB(240)
       COMMON /ENABLE/ CRITER,IREAD,IWRITE,JUPDAT,LOOP
       COMMON /SIZE/ NLESS1,NTOTAL,NLESX2,NTOTX2
       READ(5,1)PARM
       IREAD=PARM(1)
       IWRITE=PARM(2)
       CRITER=PARM(3)
       JUPDAT=PARM(4)
       LOOP=PARM(5)
       WRITE(IWRITE,5)
       CALL INTIAL
       CALL DINPUT
       CALL NEWTON(OK)
       CALL RESULT(OK)
       STOP
 1     FORMAT(5F10.5)
 5     FORMAT('1',T25,'CONVENTIONAL  LOAD  FLOW  PROGRAM  ',
      +'USING NEWTON-RAPHSON  ALGORITHM'// T28,
      +'WITH L-U DECOMPOSITION OF THE JACOBIAN',
      +' AND SPARSITY PROGRAMMING')
       END
       SUBROUTINE INTIAL
C
C
C      THIS SUBROUTINE INITIALISES VARIABLES
C
       IMPLICIT REAL*8 (A-H,O-Z), INTEGER*2 (I-N)
       COMMON /NETWOK/ DIAYMR(120),DIAYMI(120),DATAYR(200),
      +DATAYI(200),
      +DATALN(200),LKSTYM(120),JCOLYM(400),LINKYM(400),
      +KONECT(120),
      +LORDER(120),NORDER(120),LINE,NFAULT
       DO 1 I=1,120
       DIAYMR(I)=0.0
       DIAYMI(I)=0.0
       LKSTYM(I)=0
 1     KONECT(I)=0
       RETURN
       END
       SUBROUTINE DINPUT
C
C
C
```

```
C      THIS SUBROUTINE READS IN THE LINE DATA AND BUS DATA
C      THEN DOES A RE-NUMBERING OF THE BUSES ACCORDING TO
C      THIS :
C
C      PV BUSES ARE GIVEN THE FIRST NUMBERS IN THE ORDER THEY
C      ARE ENTERED THEN PQ BUSES ARE NUMBERED INASCENDING
C      ORDER OF THE NUMBER OF LINES JOINING IT
C      THE SLACK BUS IS NUMBERED LAST
C
       IMPLICIT REAL*8 (A-H,O-Z), INTEGER*2 (I-N)
       INTEGER*4 IREAD,IWRITE
       REAL*4 BUFLIN(5),BUFNOD(9),
      +NODE1,NODE2,RPU,XLPU,YCPU,NODE,TYPE,
      +VM,PG,QG,PD,QD,VMG,ANGLE
       COMMON /LOADFL/ VREAL(120),VIMAG(120),VMAGSQ(120),
      +CREAL(120),
      +CIMAG(120),CMAGLN(200),REALG(120),REACTG(120),
      +REALD(120),REACTD(120),MODBUS(120),NREF,NOGEN
       COMMON /NETWOK/ DIAYMR(120),DIAYMI(120),
      +DATAYR(200),DATAYI(200),
      +DATALN(200),LKSTYM(120),JCOLYM(400),LINKYM(400),
      +KONECT(120),
      +LORDER(120),NORDER(120),LINE,NFAULT
       COMMON /LUSOLV/
      +DELTAX(240),ERRORZ(240),DELTAG(120),DELTAQ(120),
      +DIAGUT(240),DATAUT(500),DATALT(500),LKSTUT(240),
      +JROWUT(500),
      +LINKUT(500),IRSTUT(240),JCOLUT(500),IRSTLT(240),
      +JCOLLT(500),
      +JCOLJB(240)
       COMMON /ENABLE/ CRITER,IREAD,IWRITE,JUPDAT,LOOP
       COMMON /SIZE/ NLESS1,NTOTAL,NLESX2,NTOTX2
       COMMON /CONTIG/ NODE1,NODE2,RPU,XLPU,YCPU
       COMMON /CONTIN/ NODE,TYPE,VM,PG,QG,PD,QD,VMG,ANGLE
       COMPLEX*16 REACTN,IMAG/(0.0D0,1.0D0)/
       EQUIVALENCE (BUFLIN,NODE1),(BUFNOD,NODE)
C
C      READING IN THE LINE DATA AND CALCULATING THE Y-MATRIX
C      ONLY NON-ZERO ELEMENTS OF THE Y-MATRIX ARE STORED
C
       NFAULT=-1
       LINE=0
       LIST=0
       WRITE(IWRITE,1111)
       WRITE(IWRITE,50)
10     CONTINUE
          READ(IREAD,51)BUFLIN
          IF(NODE1.EQ.0.0)GO TO 20
          WRITE(IWRITE,52)BUFLIN
          LINE=LINE+1
          LIST=LIST+1
```

```
         N2=NODE2
         N1=NODE1
         KONECT(N1)=KONECT(N1)+1
         KONECT(N2)=KONECT(N2)+1
         REACTN=1D0/(RPU+XLPU*IMAG)
         SUSCEP=-REACTN*IMAG
         DIAYMR(N1)=DIAYMR(N1)+REACTN
         DIAYMR(N2)=DIAYMR(N2)+REACTN
         DIAYMI(N1)=DIAYMI(N1)+SUSCEP+YCPU
         DIAYMI(N2)=DIAYMI(N2)+SUSCEP+YCPU
         DATAYR(LINE)=-REACTN
         DATAYI(LINE)=-SUSCEP
         DATALN(LINE)=YCPU
         JCOLYM(LIST)=N2
         LINKYM(LIST)=LKSTYM(N1)
         LKSTYM(N1)=LIST
         LIST=LIST+1
         JCOLYM(LIST)=N1
         LINKYM(LIST)=LKSTYM(N2)
         LKSTYM(N2)=LIST
         GO TO 10
20    CONTINUE
      WRITE(IWRITE,55)LINE
C
C     READING IN THE BUS DATA
C
      WRITE(IWRITE,1111)
      NTOTAL=0
      NOGEN=0
      DTORAD=3.1415926D0/180.0D0
      WRITE(IWRITE,60)
30    CONTINUE
         READ(IREAD,61)BUFNOD
         IF(NODE.EQ.0.0)GO TO 40
         WRITE(IWRITE,62)BUFNOD
         NTOTAL=NTOTAL+1
         N=NODE
         MODBUS(N)=TYPE
         VREAL(N)=VMG*DCOS(ANGLE*DTORAD)
         VIMAG(N)=VMG*DSIN(ANGLE*DTORAD)
         REALG(N)=PG
         REALD(N)=PD
         REACTG(N)=QG
         REACTD(N)=QD
         DELTAG(N)=PG-PD
         DELTAQ(N)=QG-QD
         IF(TYPE.EQ.1.0)GO TO 30
           VMAGSQ(N)=VM**2
           KONECT(N)=0
           NOGEN=NOGEN+1
           IF(TYPE.EQ.3.0)NREF=N
```

```
            GO TO 30
  40    CONTINUE
        READ(IREAD,51,END=41)NFAULT
        DIAYMR(NFAULT)=99.
  41    CONTINUE
        NLESS1=NTOTAL-1
        NLESX2=NLESS1*2
        NTOTX2=NTOTAL*2
        NOGEN=NOGEN-1
        KONECT(NREF)=100
C
C       BUS RE-NUMBERING
C
        DO 45 I=1,NTOTAL
  45    NORDER(I)=I
  65    INTERC=0
           DO 70 I=2,NTOTAL
           IF(KONECT(I-1).LE.KONECT(I))GO TO 70
             L=KONECT(I)
             KONECT(I)=KONECT(I-1)
             KONECT(I-1)=L
             L=NORDER(I)
             NORDER(I)=NORDER(I-1)
             NORDER(I-1)=L
             INTERC=1
  70       CONTINUE
           IF(INTERC.NE.0)GO TO 65
        DO 80 I=1,NTOTAL
  80    LORDER(NORDER(I))=I
        RETURN
  50    FORMAT(' LINE DATA'/1X,9('=')///T8,
       +'BUS NO. JOINS BUS NO.',T36,
       +'R  P.U.',T50,'XL  P.U.',T64,
       +'YSH  P.U.'/'+',T8,7('_'),T22,7('_'),
       +T35,'___',T49,'____',T63,'_____'///)
  51    FORMAT(8F10.5)
  52    FORMAT(8X,F4.0,10X,F4.0,1X,3F14.4/)
  55    FORMAT(//// T22,'THERE ARE ',I4,
       +' LINES IN THE SYSTEM')
  60    FORMAT(' BUS DATA'/1X,8('=')///T24,
       +'VOLTAGE',T37,'GENERATION',T57,
       +'LOAD',T71,'STARTING VOLTAGE'//T8,'NUMBER',T16,
       +'TYPE',T23,
       +'MAGNITUDE',T35,
       +'REAL',T41,'REACTIVE',T54,'REAL',T60,'REACTIVE',
       +T71,'MAGNITUDE',T82,'ANGLE'/'+',T8,6('_'),T16,
       +'____',T23,9('_'),
       +T35,'____',T41,8('_'),T54,'____',T60,8('_'),T71,9('_')
       +,T82,5('_')///)
  61    FORMAT(16F5.2)
  62    FORMAT(8X,F4.0,T17,F2.0,T25,F4.2,
```

```
     +T34,F6.3,T42,F6.3,T53,F6.3,
     +T61,F6.3,T71,F6.2,T80,F6.2/)
1111  FORMAT(//////////)
      END
      SUBROUTINE NEWTON(OK)
C
C
C
C     THIS SUBROUTINE PERFORMS THE NEWTON RAPHSON ITERATIONS
C
C
C     IT RETURNS A LOGICAL*1 VARIABLE OF VALUE .TRUE. WHEN
C     THE ROUTINE WAS SUCCESSFULLY COMPLETED
C     OTHERWISE THE RETURNED VARIABLE IS .FALSE.
C     UNSUCCESSFUL COMPLETION IS WHEN THE REQUIRED ACCURACY
C     IS NOT ATTAINED IN THE MAXIMUM ALLOWED NUMBER OF
C     ITERATIONS
C
C
      IMPLICIT REAL*8 (A-H,O-Z), INTEGER*2 (I-N)
      INTEGER*4 IREAD,IWRITE
      LOGICAL*1 OK
      COMMON /LOADFL/ VREAL(120),VIMAG(120),VMAGSQ(120),
     +CREAL(120),
     +CIMAG(120),CMAGLN(200),REALG(120),REACTG(120),
     +REALD(120),REACTD(120),MODBUS(120),NREF,NOGEN
      COMMON /NETWOK/ DIAYMR(120),DIAYMI(120),
     +DATAYR(200),DATAYI(200),
     +DATALN(200),LKSTYM(120),JCOLYM(400),LINKYM(400),
     +KONECT(120),LORDER(120),NORDER(120),LINE,NFAULT
      COMMON /LUSOLV/
     +DELTAX(240),ERRORZ(240),DELTAG(120),DELTAQ(120),
     +DIAGUT(240),DATAUT(500),DATALT(500),LKSTUT(240),
     +JROWUT(500),
     +LINKUT(500),IRSTUT(240),JCOLUT(500),IRSTLT(240),
     +JCOLLT(500),JCOLJB(240)
      COMMON /ENABLE/ CRITER,IREAD,IWRITE,JUPDAT,LOOP
      COMMON /SIZE/ NLESS1,NTOTAL,NLESX2,NTOTX2
      KOUNT=0
100   CONTINUE
C
C        ** CALCULATING CURRENT INJECTIONS
         DO 200 M=1,NTOTAL
            I=NORDER(M)
            CREAL(M)=DIAYMR(I)*VREAL(I)-DIAYMI(I)*VIMAG(I)
            CIMAG(M)=DIAYMI(I)*VREAL(I)+DIAYMR(I)*VIMAG(I)
            L=LKSTYM(I)
150         CONTINUE
               KDATA=(L+1)/2
               J=JCOLYM(L)
               CREAL(M)=CREAL(M)+DATAYR(KDATA)*VREAL(J)
```

```
     +                                    -DATAYI(KDATA)*VIMAG(J)
                CIMAG(M)=CIMAG(M)+DATAYR(KDATA)*VIMAG(J)+
     +                                    DATAYI(KDATA)*VREAL(J)
                L=LINKYM(L)
                IF(L.NE.0)GO TO 150
200       CONTINUE
C
C         ** EVALUATING THE MISMATCHES
          IF (NOGEN.EQ.0)GO TO 410
          DO 400 I=1,NOGEN
             J=NORDER(I)
             ERRORZ(I)=-VMAGSQ(J)+VREAL(J)**2+VIMAG(J)**2
             ERRORZ(I+NLESS1)=-DELTAG(J)+VREAL(J)*CREAL(I)
     +                                  +VIMAG(J)*CIMAG(I)
400       CONTINUE
410       CONTINUE
          IF(NOGEN.EQ.NLESS1)GO TO 510
          M=NOGEN+1
          DO 500 I=M,NLESS1
             J=NORDER(I)
             ERRORZ(I)=-DELTAQ(J)-(VREAL(J)*CIMAG(I)
     +                                     -VIMAG(J)*CREAL(I))
             ERRORZ(I+NLESS1)=-DELTAG(J)+(VREAL(J)*CREAL(I)+
     +                             VIMAG(J)*CIMAG(I))
       IF(J.EQ.NFAULT)ERRORZ(I)=0.0
       IF(J.EQ.NFAULT)ERRORZ(I+NLESS1)=0.0
500       CONTINUE
510       CONTINUE
C
C         ** CHECKING AGAINST CONVERGENCE CRITERIA
          DO 600 I=1,NLESX2
             IF(DABS(ERRORZ(I)).GT.CRITER)GO TO 700
600       CONTINUE
C
C         ** ALL MISMATCHES ARE LESS THAN THE CRITERIA GIVEN
          WRITE(IWRITE,550)CRITER,KOUNT
          OK=.TRUE.
          RETURN
C
C         ** MORE ITERATIONS REQUIRED
C
700       CONTINUE
          IF(KOUNT.LT.LOOP)GO TO 710
          WRITE(IWRITE,560)CRITER,LOOP
          OK=.FALSE.
          RETURN
C
C         ** DETERMINE WHETHER TO UPDATE JACOBIAN MATRIX
710       CONTINUE
          IF(KOUNT.EQ.0)GO TO 730
          IF(JUPDAT.EQ.0)GO TO 750
```

```
          IF(KOUNT/JUPDAT.EQ.(KOUNT-1)/JUPDAT)GO TO 750
730       CONTINUE
          CALL JACOB
750       CONTINUE
          CALL BACKSB(NLESX2)
          DO 800 I=1,NLESS1
             J=NORDER(I)
             VREAL(J)=VREAL(J)+DELTAX(I)
             VIMAG(J)=VIMAG(J)+DELTAX(I+NLESS1)
       WRITE(IWRITE,660)J,VREAL(J),VIMAG(J)
660    FORMAT(// '    VOLTAGE AT BUS  ',I3 ,':   ',F10.5,
     +'  +  J',F10.5)
800       CONTINUE
          KOUNT=KOUNT+1
       GO TO 100
550    FORMAT('1',T11,'CONVERGES TO WITHIN ',F8.5,' FOR THE',
     + MAXIMUM MISMATCH IN ',I3,' ITERATIONS' /////)
560    FORMAT('1',T11,'FAILS TO CONVERGE TO WITHIN ',F8.5,
     +' FOR THE MAXIMUM',
     + MISMATCH IN ',I3,' ITERATIONS' /////)
       END
       SUBROUTINE JACOB
C
C
C
C      THIS SUBROUTINE EVALUATES THE JACOBIAN MATRIX
C
C
C      THE JACOBIAN MATRIX IS NOT STORED BUT AS SOON AS ONE
C      ROW IS CALCULATED IT IS DECOMPOSED INTO THE
C      CORRESPONDING ROWS OF THE LOWER AND UPPER TRIANGULAR
C      MATRICES
C
C
       IMPLICIT REAL*8 (A-H,O-Z), INTEGER*2 (I-N)
       COMMON /LOADFL/ VREAL(120),VIMAG(120),VMAGSQ(120),
     +CREAL(120),
     +CIMAG(120),CMAGLN(200),REALG(120),REACTG(120),
     +REALD(120),REACTD(120),MODBUS(120),NREF,NOGEN
       COMMON /NETWOK/ DIAYMR(120),DIAYMI(120),
     +DATAYR(200),DATAYI(200),
     +DATALN(200),LKSTYM(120),JCOLYM(400),LINKYM(400),
     +KONECT(120),
     +LORDER(120),NORDER(120),LINE,NFAULT
       COMMON /LUSOLV/ DATAJB(240),ERRORZ(240),
     +DELTAG(120),DELTAQ(120),
     +DIAGUT(240),DATAUT(500),DATALT(500),LKSTUT(240),
     +JROWUT(500),
     +LINKUT(500),IRSTUT(240),JCOLUT(500),IRSTLT(240),
     +JCOLLT(500),JCOLJB(240)
       COMMON /SIZE/ NLESS1,NTOTAL,NLESX2,NTOTX2
```

```
      DO 170 J=1,NLESS1
      I=NORDER(J)
      DO 179 J0=1,NTOTX2
         DATAJB(J0)=0.0
179      JCOLJB(J0)=0
      J0=1
      IF(MODBUS(I).NE.2)GO TO 175
         DATAJB(J)=-2.0*VREAL(I)
         JCOLJB(J0)=J+NLESS1
         DATAJB(J+NLESS1)=-2.0*VIMAG(I)
      GO TO 178
175      DATAJB(J)=-VIMAG(I)*DIAYMR(I)
     +                    +VREAL(I)*DIAYMI(I)+CIMAG(J)
         L=LKSTYM(I)
171      CONTINUE
            M=JCOLYM(L)
            IF(M.EQ.NREF)GO TO 177
            M=LORDER(M)
            JCOLJB(J0)=M
            J0=J0+1
            KD=(L+1)/2
            DATAJB(M)=-VIMAG(I)*DATAYR(KD)
     +                          +VREAL(I)*DATAYI(KD)
            JCOLJB(J0)=M+NLESS1
            J0=J0+1
            DATAJB(M+NLESS1)=VIMAG(I)*DATAYI(KD)+VREAL(I)*DATAYR(KD)
177         L=LINKYM(L)
            IF(L.NE.0)GO TO 171
        JCOLJB(J0)=J+NLESS1
        DATAJB(J+NLESS1)=VIMAG(I)*DIAYMI(I)
     +                         +VREAL(I)*DIAYMR(I)-CREAL(J)
178   CALL LUNSYM(J,NLESX2)
170   CONTINUE
      DO 160 J=NTOTAL,NLESX2
      I=NORDER(J-NLESS1)
      DO 163 J0=1,NTOTX2
         DATAJB(J0)=0.0
163      JCOLJB(J0)=0
      J0=1
      DATAJB(J)=-VIMAG(I)*DIAYMR(I)
     +               +VREAL(I)*DIAYMI(I)-CIMAG(J-NLESS1)
      L=LKSTYM(I)
161      CONTINUE
         M=JCOLYM(L)
         IF(M.EQ.NREF)GO TO 166
         M=LORDER(M)
         JCOLJB(J0)=M
         J0=J0+1
         KD=(L+1)/2
         DATAJB(M)=-VREAL(I)*DATAYR(KD)-VIMAG(I)*DATAYI(KD)
         JCOLJB(J0)=M+NLESS1
```

```
            J0=J0+1
            DATAJB(M+NLESS1)=VREAL(I)*DATAYI(KD)
         +                                  -VIMAG(I)*DATAYR(KD)
166         L=LINKYM(L)
         IF(L.NE.0)GO TO 161
         JCOLJB(J0)=J-NLESS1
         DATAJB(J-NLESS1)=-VREAL(I)*DIAYMR(I)
        +                     -VIMAG(I)*DIAYMI(I)
        +                     -CREAL(J-NLESS1)
         CALL LUNSYM(J,NLESX2)
160      CONTINUE
         RETURN
         END
         SUBROUTINE LUNSYM(I,N)
C
C
C
C        THIS SUBROUTINE TRANSFORMS A GIVEN ROW OF A MATRIX
C        INTO THE CORRESPONDING ROWS OF A LOWER TRIANGULAR AND
C        AN UPPER TRIANGULAR MATRIX.  ONLY NON-ZERO ELEMENTS
C        ARE STORED.
C
C        THE INPUT VARIABLES I GIVES WHAT ROW OF THE MATRIX IT
C        IS TO BE DECOMPOSED WHILE N GIVES THE ORDER OF THE
C        MATRIX
C
C
         IMPLICIT REAL*8 (A-H,O-Z), INTEGER*2 (I-N)
         COMMON /LUSOLV/ DATAJB(240),ERRORZ(240),
        +DELTAG(120),DELTAQ(120),
        +DIAGUT(240),DATAUT(500),DATALT(500),LKSTUT(240),
        +JROWUT(500),
        +LINKUT(500),IRSTUT(240),JCOLUT(500),IRSTLT(240),
        +JCOLLT(500),JCOLJB(240)
         IF(I.NE.1)GO TO 22
C
C        ** INITIALIZATION AND CALCULATION OF FIRST ROW OF
C           UPPER TRIANGULAR
         KUT=1
         KLT=1
         IRSTUT(1)=1
         IRSTLT(1)=1
         DO 1 M=1,N
1        LKSTUT(M)=0
         DIAGUT(1)=DATAJB(1)
         J0=1
         L=JCOLJB(J0)
10       IF(L.EQ.0)GO TO 20
            IF(DATAJB(L).EQ.0.0)GO TO 15
               DATAUT(KUT)=DATAJB(L)
               JROWUT(KUT)=1
```

```
               JCOLUT(KUT)=L
               LINKUT(KUT)=LKSTUT(L)
               LKSTUT(L)=KUT
               KUT=KUT+1
15          CONTINUE
            J0=J0+1
            L=JCOLJB(J0)
         GO TO 10
20       RETURN
C
C        **DECOMPOSITION OF ROWS OTHER THAN THE FIRST
22       CONTINUE
         J0=1
         IRSTUT(I)=KUT
         IRSTLT(I)=KLT
         IR=IRSTLT(I)
C
C        ** SEEKING COLUMN ONE ENTRIES OF JACOBIAN MATRIX
         L=JCOLJB(J0)
130      IF(L.EQ.0)GO TO 110
            IF(L.EQ.1)GO TO 120
            J0=J0+1
            L=JCOLJB(J0)
         GO TO 130
C
C        ** EVALUATING ELEMENT OF COLUMN ONE OF LOWER
C           TRIANGULAR MATRIX
120      JCOLLT(KLT)=1
         DATALT(KLT)=DATAJB(L)/DIAGUT(1)
         KLT=KLT+1
C
C        ** IF THIS IS SECOND ROW  NO MORE LOWER TRIANGULAR
C           MATRIX ENTRIES
110      IF(I.EQ.2)GO TO 140
         I1=I-1
         DO 200 J=2,I1
         J0=1
         DATALT(KLT)=0.0
C
C        ** SEEKING NON-ZERO ELEMENT IN CORRESPONDING POSITION
C           IN JACOBIAN
         L=JCOLJB(J0)
220      IF(L.EQ.0)GO TO 230
            IF(L.EQ.J)GO TO 210
            J0=J0+1
            L=JCOLJB(J0)
         GO TO 220
210      DATALT(KLT)=DATAJB(L)
230      K1=KLT-1
C
C        ** IF THERE ARE NO PREVIOUS ENTRIES IN THIS ROW OF
```

```
C          THE LOWER TRAIN MATRIX NO FURTHER PROCESSING IS
C          NECESSARY FOR THIS ELEMENT
C
      IF(IR.GT.K1)GO TO 240
C
C     ** SCANNING THROUGH LIST OF ELEMENTS IN COLUMN J OF
C          UPPER TRIANGULAR MATRIX TO MATCH THE CORRESPONDING
C          ENTRY IN THE LOWER TRIANGULAR
      L=LKSTUT(J)
      MM=KLT
      DO 250 M=IR,K1
         MM=MM-1
270      IF(L.EQ.0)GO TO 250
         IF(JROWUT(L).LT.JCOLLT(MM))GO TO 250
            IF(JROWUT(L).EQ.JCOLLT(MM))GO TO 260
            L=LINKUT(L)
         GO TO 270
260      DATALT(KLT)=DATALT(KLT)-DATAUT(L)*DATALT(MM)
250   CONTINUE
C
C     ** IF ELEMENT IS ZERO DO NOT STORE INTO LIST
240   IF(DATALT(KLT).EQ.0.0)GO TO 200
         JCOLLT(KLT)=J
         DATALT(KLT)=DATALT(KLT)/DIAGUT(J)
         KLT=KLT+1
200   CONTINUE
C
C     ** CALCULATING THE DIAGONAL ELEMENT OF UPPER
C          TRIANGULAR MATRIX
140   DIAGUT(I)=DATAJB(I)
      K1=KLT-1
      IF(IR.GT.K1)GO TO 340
         L=LKSTUT(I)
         MM=KLT
         DO 300 J=IR,K1
            MM=MM-1
330         IF(L.EQ.0)GO TO 300
            IF(JROWUT(L).LT.JCOLLT(MM))GO TO 300
               IF(JROWUT(L).EQ.JCOLLT(MM))GO TO 320
               L=LINKUT(L)
            GO TO 330
320         DIAGUT(I)=DIAGUT(I)-DATALT(MM)*DATAUT(L)
300      CONTINUE
340   CONTINUE
C
C     ** IF IT IS THE LAST ROW  THERE IS NO NON-DIAGONAL
C          ELEMENT IN UPPE
      IF(I.EQ.N)GO TO 100
C
C     ** EVALUATING ELEMENTS IN THE UPPER TRIANGULAR MATRIX
C          EXCLUDING DIAGONAL
```

```
      I1=I+1
      DO 400 J=I1,N
      J0=1
      DATAUT(KUT)=0.0
      L=JCOLJB(J0)
430   IF(L.EQ.0)GO TO 410
         IF(L.EQ.J)GO TO 420
            J0=J0+1
            L=JCOLJB(J0)
         GO TO 430
420      DATAUT(KUT)=DATAJB(L)
410   IF(IR.GT.K1)GO TO 440
      L=LKSTUT(J)
      MM=KLT
      DO 450 M=IR,K1
         MM=MM-1
480      IF(L.EQ.0)GO TO 450
         IF(JROWUT(L).LT.JCOLLT(MM))GO TO 450
            IF(JROWUT(L).EQ.JCOLLT(MM))GO TO 470
            L=LINKUT(L)
         GO TO 480
470      DATAUT(KUT)=DATAUT(KUT)-DATALT(MM)*DATAUT(L)
450   CONTINUE
440   IF(DATAUT(KUT).EQ.0.0)GO TO 400
         JROWUT(KUT)=I
         LINKUT(KUT)=LKSTUT(J)
         LKSTUT(J)=KUT
         JCOLUT(KUT)=J
         KUT=KUT+1
400   CONTINUE
      RETURN
100   CONTINUE
      IRSTLT(N+1)=KLT
      IRSTUT(N+1)=KUT
      RETURN
      END
      SUBROUTINE BACKSB(N)
C
C
C
C     THIS SUBROUTINE DOES A FORWARD THEN A BACKWARD
C     SUBSTITUTION WITH THE GIVEN LOWER AND UPPER TRIANGULAR
C     MATRICES RESPECTIVELY.
C
C
C
C     THE INPUT VARIABLE N IS THE NUMBER OF ROWS IN THE
C     MATRIX
C
C
      IMPLICIT REAL*8 (A-H,O-Z), INTEGER*2 (I-N)
```

```
      COMMON /LUSOLV/
     +DELTAX(240),ERRORZ(240),DELTAG(120),DELTAQ(120),
     +DIAGUT(240),DATAUT(500),DATALT(500),LKSTUT(240),
     +JROWUT(500),
     +LINKUT(500),IRSTUT(240),JCOLUT(500),IRSTLT(240)
     +JCOLLT(500),JCOLJB(240)
      IEND=IRSTLT(2)-1
      DO 600 I=1,N
600   DELTAX(I)=ERRORZ(I)
      DO 700 I=2,N
         ISTART=IEND+1
         IEND=IRSTLT(I+1)-1
         IF(IEND.LT.ISTART)GO TO 700
         DO 750 J=ISTART,IEND
            M=JCOLLT(J)
            DELTAX(I)=DELTAX(I)-DATALT(J)*DELTAX(M)
750      CONTINUE
700   CONTINUE
      ISTART=IRSTUT(N+1)
      DO 800 I=1,N
         IR=N-I+1
         IEND=ISTART-1
         ISTART=IRSTUT(IR)
         IF(IEND.LT.ISTART)GO TO 800
         DO 850 J=ISTART,IEND
            M=JCOLUT(J)
            DELTAX(IR)=DELTAX(IR)-DATAUT(J)*DELTAX(M)
850      CONTINUE
800   DELTAX(IR)=DELTAX(IR)/DIAGUT(IR)
      RETURN
      END
      SUBROUTINE RESULT(OK)
C
C
C
C     THIS SUBROUTINE CALCULATES THE POWER INJECTIONS, THE
C     POWER FLOWS AND THE TOTAL TRANSMISSION LOSSES OF A
C     SYSTEM GIVEN THE NODAL VOLTAGES
C
C
C     IF THE INPUT VARIABLE IS .FALSE. IT PRINTS A WARNING
C     MESSAGE THAT THE NODAL VOLTAGES ARE NOT UP TO THE
C     SUFFICIENT ACCURACY
C
C
      IMPLICIT REAL*8 (A-H,O-Z), INTEGER*2 (I-N)
      INTEGER*4 IREAD,IWRITE
      COMMON /LOADFL/ VREAL(120),VIMAG(120),VMAGSQ(120),
     +CREAL(120),
     +CIMAG(120),CMAGLN(200),REALG(120),REACTG(120),
     +REALD(120),REACTD(120),MODBUS(120),NREF,NOGEN
```

```
      COMMON /NETWOK/ DIAYMR(120),DIAYMI(120),
     +DATAYR(200),DATAYI(200),
     +DATALN(200),LKSTYM(120),JCOLYM(400),LINKYM(400),
     +KONECT(120),LORDER(120),NORDER(120),LINE,NFAULT
      COMMON /LUSOLV/
     +DELTAX(240),ERRORZ(240),DELTAG(120),DELTAQ(120),
     +DIAGUT(240),DATAUT(500),DATALT(500),LKSTUT(240),
     +JROWUT(500),
     +LINKUT(500),IRSTUT(240),JCOLUT(500),IRSTLT(240),
     +JCOLLT(500),JCOLJB(240)
      COMMON /ENABLE/ CRITER,IREAD,IWRITE,JUPDAT,LOOP
      COMMON /SIZE/ NLESS1,NTOTAL,NLESX2,NTOTX2
      LOGICAL*1 OK
      IF(.NOT.OK)WRITE(IWRITE,500)
      WRITE(IWRITE,520)
      PLOSS=0.0
      RTODEG=180D0/3.1415926D0
      DO 1000 I=1,NTOTAL
         VMAG=DSQRT(VREAL(I)**2+VIMAG(I)**2)
         ANGLE=DATAN2(VIMAG(I),VREAL(I))*RTODEG
         J=LORDER(I)
         IF(MODBUS(I).EQ.1)GO TO 300
            REACTG(I)=VIMAG(I)*CREAL(J)
     +                       -VREAL(I)*CIMAG(J)+REACTD(I)
            IF(MODBUS(I).NE.3)GO TO 300
               REALG(I)=REALD(I)+VREAL(NREF)*CREAL(NTOTAL)
               WRITE(IWRITE,550)I,VMAG,ANGLE,REALG(I),
     +         REACTG(I),REALD(I),REACTD(I)
               GO TO 350
300      CONTINUE
         WRITE(IWRITE,550)I,VMAG,ANGLE,REALG(I),REACTG(I),
     +   REALD(I),REACTD(I),ERRORZ(J),ERRORZ(J+NLESS1)
350      CONTINUE
         PLOSS=PLOSS+REALG(I)-REALD(I)
         L=LKSTYM(I)
400      CONTINUE
           M=JCOLYM(L)
           KDATA=(L+1)/2
           VOLTRL=VREAL(I)-VREAL(M)
           VOLTIM=VIMAG(I)-VIMAG(M)
           CURREL=-DATAYR(KDATA)*VOLTRL+DATAYI(KDATA)*VOLTIM
           CURIMG=-DATAYR(KDATA)*VOLTIM-DATAYI(KDATA)*VOLTRL
           SHUNT=-DATALN(KDATA)*VMAG**2
           REAFLO=VREAL(I)*CURREL+VIMAG(I)*CURIMG
           REACTV=VIMAG(I)*CURREL-VREAL(I)*CURIMG+SHUNT
           WRITE(IWRITE,555)M,REAFLO,REACTV
           L=LINKYM(L)
           IF(L.NE.0)GO TO 400
1000  CONTINUE
      WRITE(IWRITE,559)PLOSS
      RETURN
```

```
500    FORMAT(T14,'THE FOLLOWING RESULTS ARE CALCULATED ',
      +'BASED ON THE YET TO CONVERGE VOLTAGES' /////)
520    FORMAT(/ T14,'V O L T A G E',T35,
      +'GENERATION',T56,'DEMAND',
      +T73,'MISMATCH' // T11,'MAGNITUDE   ANGLE',
      +T33,'REAL   REACTIVE',
      +T52,'REAL   REACTIVE',T71,'Q/^V^  POWER' / )
550    FORMAT(// ' BUS ',I3,F10.3,F8.2,3(3X,2F8.3)/)
555    FORMAT(7X,'TO BUS ',I3,T30,2F8.3)
559    FORMAT(/////T20,'TOTAL SYSTEM LOSS = ',F8.3 / '1')
       END
C
C
C
C
C
C
C
C
$DATA
5.        10.       0.0002    1.        4.
1.        2.        0.02      0.06      0.03
1.        3.        0.08      0.24      0.025
2.        3.        0.06      0.18      0.02
2.        4.        0.06      0.18      0.02
2.        5.        0.04      0.12      0.015
3.        4.        0.01      0.03      0.01
4.        5.        0.08      0.24      0.025

1.   3.   1.06 0.              1.06
2.   2.   1.05 0.692      0.2 0.1  1.
3.   2.   1.04 0.527      0.450.15 1.
4.   1.   1.              0.4 0.05 1.
5.   1.   1.              0.6 0.1  1.

$
```