IMPERIAL COLLEGE OF SCIENCE AND TECHNOLOGY

(University of London)

Department of Electrical Engineering

A Thesis on

The Determination of the Performance of Protection Applied to Industrial Distribution Systems Containing Large Induction Motor Loads and Subject to Asymmetric Faults.

by

D. J. Davey, B.Sc.

Presented for the degree of Master of Philosophy.

1976

ACKNOWLEDGEMENTS

The work presented in this report was carried out under the supervision of Dr. B.J. Cory, D.Sc., F.I.E.E., C.Eng., of the Department of Electrical Engineering, Imperial College of Science and Technology, London.

I wish to thank Dr. Cory for his helpful guidance, constant encouragement and forebearance.

Table of Contents

A List of the Principal Symbols Used

$I_k = I_{pk} + jI_{qk}$	injected nodal current at node K
$V_k = V_{pk} + jV_{qk}$	voltage at node K
V_p	positive sequence voltage
V_n	negative sequence voltage
V_z	zero sequence voltage
$Y_{kk} = G_{kk} + jB_{kk}$	the total admittance oonnected to node K
$Y_{km} = G_{km} + jB_{km}$	the admittance connected between nodes K and M
$S_k = P_k + jQ_k$	the injected nodal apparent power at node K
$S_k = V_k.I_k^*$	
X''	subtransient reactance
X'	transient reactance
X_s	synchronous reactance
T''	subtransient time constant
T'	transient time constant
T_a	armature time constant
P.S.N.	positive sequence network
N.S.N.	negative sequence network
Z.S.N.	zero sequence network
P.S.N.A.M.	positive sequence nodal admittance matrix
N.S.N.A.M.	negative sequence nodal admittance matrix
Z.S.N.A.M.	zero sequence nodal admittance matrix
Z_p	positive sequence impedance
Z_n	negative sequence impedance
Z_o	zero sequence impedance
E_{max}	maximum value of generated emf.

Table of Contents - Chapter 1

Page

Table of Contents - Chapter 2 | Page

Table of Contents - Chapter 3

Table of Contents - Chapter 4 | Page

Table of Contents - Chapter 5 Page

Abstract

The grading and operation of protective gear is usually based on the normal operating conditions that apply when a system is commissioned. For small, and or simple systems the effect of outage and reinforcement can easily be accommodated. Large systems, however, present a problem of greater magnitude; reinforcements, increasing fault levels and changes in the 'normal' running arrangement can result in the basic grading pattern being lost. This work provides a possible aid to overcoming such problems for industrial type distribution systems, since the analyses include the effects of induction motor loads, which have in the past, not always correctly, been neglected as a source of fault current.

Chapter 2 discusses the adopted approach to system analysis, which uses the Newton -Raphson technique aided by the Epsilon Algorithm to solve the network equations. The types of analysis available from the described program are:

a) balanced load flow
b) three phase symmetrical short circuit
c) asymmetric short circuit - steady and non-steady state.

Since the non-steady state analysis is derived by interpolating the results of three load flows, the number of machines that can be included in the analysis is limited only by the amount of computer storage available.

Chapter 3 discusses relay characteristics and the methods used to synthesise these characteristics in the relay subroutines. As the non-steady state analysis is derived through interpolative methods special curve fitting techniques have been developed for this purpose, and these are also discussed in this chapter.

Chapter 4 gives an outline of the working of the program and the conclusions that have been made as a result of this work.

There are two appendices, chapters 5 and 6; chapter 5 describes how the program may be used and details the facilities that it offers. Chapter 6 has been written so that other workers may gain a detailed understanding of the program and its methods of operation. This appendix also contains a full program listing.

CHAPTER 1

INTRODUCTION

1.1 Objectives

There have been three main objectives in performing this work:

a) to determine the non steady state behaviour of an industrial power system containing induction motor loads, by performing a minimum number of load flows.

b) to use the results of a) to compute the dynamic behaviour of the system protection.

c) to produce a computer program satisfying a) and b) above, which may be used by practising engineers to check the action of protective systems following system changes.

1.2 Previous Work

Previous work has led to the development of programs which fall into two main categories:

1. The determination of relay and circuit breaker operation over an extended period of network time. These are essentially programs for checking the sequence of operation of the system relays.

2. The determination of relay settings for a given network configuration.

The mode of operation of the two types of program is quite different since type 1 depends on some form of continuous systems analysis, and type 2 uses values of the system variables that have been computed for one instant of time only.

Alderton and Peralta[20] have designed a program (type 1) based on a transient stability analysis which uses a step by step integration technique for checking the operation of relays in power transmission systems. The I.D.M.T. relay characteristics are represented by a pair of empirical equations which give the relay operation times in terms of a fastest and slowest value - a band characteristic.

As each relay operation is detected, comparison is made with a master list which specifies the desired relay discrimination. If maloperation occurs, interactive facilities are provided so that the relay settings can be adjusted as the analysis proceeds.

Albrect[19] has designed a 'Protective Device Co-ordination Program' which is of type 2. This is a very comprehensive program which can select appropriate protective devices from a list of alternatives, as well as determining the device settings. The relay characteristic equations used in this program require 20 constants for each relay.

Begian[21] has also designed a comprehensive co-ordination program which is similar to that by Albrect, and again the relay characteristic equations are lengthy and inferior to those proposed by the author.

Graham and Watson[17] have produced a relay operation checking program which works from values of voltage and current computed for one instant of network time. The program uses tabulated relay characteristics and is not as comprehensive as those of Albrect and Begian.

1.3 Contributions to Research

This research has shown that I.D.M.T. type relay characteristics can be accurately synthesised by a minimum of coefficients by using the rational function curve fitting techniques which are developed in Chapter 3. Also, the decay of fault current in an electrical power distribution system may be simulated by interpolating the results obtained from a series of load flows.

The resulting program should prove to be a useful aid to the design, protection and operations engineers in the electricity supply industry.

CHAPTER 2

SYSTEM ANALYSIS

2.1 General

Modern electrical power systems are complex interconnected networks with load and generation centres that may be many miles apart. The design and operation of such systems is facilitated by the comprehensive analyses which have been made possible by the advent of the digital computer. When an analysis program has been developed it can be used by design and operation engineers to provide information about proposed or actual operating conditions. Design engineers will use this data to ensure that the circuit breaker ratings are adequate and that following the removal of a faulted section, the alternative system arrangement provides for a minimum of disruption.

The program developed through this work enables the following analyses to be made:

1. Balanced load flow
2. Symmetrical three phase short circuit
3. Asymmetrical short circuit
4. Protective system performance under asymmetric fault conditions.

Circuit breaker ratings for industrial distribution systems have, in the past, been determined from short-circuit level calculations which have neglected the fault contribution due to induction motor loads. Recent workers [1,2,3,4,5,6,7] have however, shown that this contribution can be of significant proportions. It is most probable therefore, that in many industrial distribution systems with a relatively large number of induction motors (or a few of large ratings) some circuit breakers are likely to be under-rated.

System analyses allowing for the fault contribution from all system machines have not been made since it has not yet been possible to formulate the complete set of differential equations which describe a generalised multimachine system. In the above analyses 2, 3 and 4 allowance is made for the contribution to the fault current made by the induction motors in the system. Since the analyses are based on load flow methods the number of machines is limited only by the size of computer store available.

2.2 The Basis of the Load Flow Analysis Program

The method used to perform the load flow analysis is based on the nodal voltage equations of the system, i.e.

$$[I] = [Y][V] \qquad (2.1)$$

where $[I]$ is the injected nodal current vector

$[V]$ is the nodal voltage vector

$[Y]$ is the nodal admittance matrix

For any given node

$$I_k = \sum_{m=1}^{N} Y_{km} \cdot V_m \qquad (2.2)$$

where N is the number of nodes, this includes the nodes within the equivalent circuits which are used to represent generatörs, induction motors and transformers (three winding).

The solution of these system equations is obtained by use of the Newton-Raphson method, which was originally used in this context by J.B. Ward and H.W. Hale[8]. The following derivation is based on their method, which has the following advantages:

1. It is easy to program
2. The convergence characteristics are good
3. A matrix inversion is not required
4. Network modifications are easily accommodated.

The system equations are non-linear, and cannot be solved explicitly. Therefore, an iterative scheme is used.

To provide the system with a voltage reference level, the voltage has to be specified at, at least, one bus-bar. This bus-bar is usually known as the swing, or slack, bus-bar. The specified data at all other bus-bars usually takes one of two forms;

1. Either the values of the real and reactive power are specified (PQ bus-bar), or

2. The voltage modulus and the value of real power are specified (PV bus-bar).

2.2.1 Equations for PQ Bus-Bars

The initial values of voltage at PQ bus-bars is assumed to be 1.0 + j0.0 p.u. This is a reasonable approximation since the voltage profile of a healthy system is nearly level.

For any bus-bar, k, we have equation 2.1.

$$I_k = \sum_{m=1}^{N} Y_{km} . V_m$$

also,

$$S_k = V_k I_k^* \qquad (2.3)$$

where S_k is the apparent injected power at bus-bar k.

For a PQ bus-bar the value of apparent power is specified, S_{ks}, therefore, until all voltages become equal to their solution value, the value of S_k computed from equation (2.3) will not be equal to S_{ks}, the specified value. However, a modified value of voltage $V_k + \delta V_k$ can be found for this bus-bar such that the computed value of apparent power is equal to the specified value, all other voltages remaining constant.

Thus (2.3) becomes

$$S_{ks} = (V_k + \delta V_k)\ (I_k + Y_{kk}\delta V_k)^* \qquad (2.4)$$

neglecting second order terms,

$$S_{ks} = V_k \, I_k^* + V_k \, Y_{kk}^* \, \delta V_k^* + I_k^* \delta V_k$$

Therefore, as $V_k I_k^* = S_k$

$$(S_{ks} - S_k) = \delta P + j\delta Q$$

$$= (V_{pk} + jV_{qk}) \; (G_{kk} - jB_{kk}) \; (\delta V_p - j\delta V_{qk}) + I_k^* \delta V_k^*$$

Hence,

$$\delta P = \delta V_{pk} \, (G_{kk} \, V_{pk} + B_{kk} \, V_{qk} + I_{pk})$$
$$+ \; \delta V_{qk} \, (G_{kk} \, V_{qk} - B_{kk} \, V_{pk} + I_{qk}) \qquad (2.5)$$

$$\delta Q = \delta V_{pk} \, (G_{kk} \, V_{qk} - B_{kk} \, V_{pk} - I_{qk})$$
$$+ \; \delta V_{qk} \, (I_{pk} - G_{kk} \, V_{pk} - B_{kk} \, V_{qk}) \qquad (2.6)$$

The pair of simultaneous equations (2.5) and (2.6) can be solved for the two components of the voltage change, δVp and δVq. The new, improved value of voltage at bus-bar k is,

$$V_k + \delta V_k = (V_{pk} + \delta V_{pk}) + j(V_{qk} + \delta V_{qk}) \qquad (2.7)$$

2.2.2 <u>Equations for PV Bus-Bars</u>

For a PV bus-bar a new value of voltage is required such that:

$$\left| V_k^{(m)} \right| = \left| V_k^{(m+1)} \right|$$

or,

$$\left| V_p^{(m)} + jV_q^{(m)} \right| = \left| V_p^{(m+1)} + jV_q^{(m+1)} \right|$$

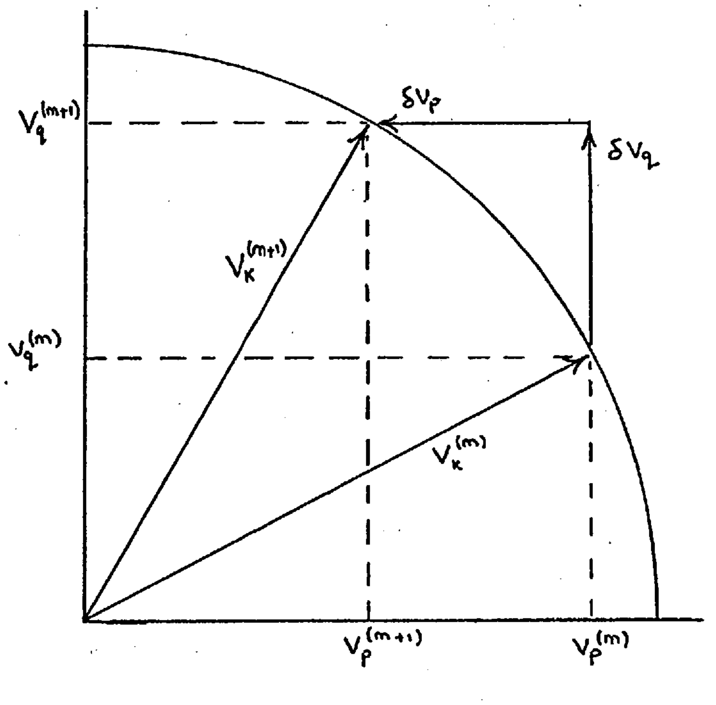


Fig. 2.1.

and

$$\left| V_p^{(m)} + \delta V_p + j(V_q^{(m)} + \delta V_q) \right| = \left| V_p^{(m+1)} + jV_q^{(m+1)} \right| \quad (2.8)$$

all values applying to bus-bar k. Therefore,

$$(V_p^{(m)} + \delta V_p)^2 + (V_q^{(m)} + \delta V_q)^2 = (V_p^{(m+1)})^2 + (V_q^{(m+1)})^2$$

expanding and neglecting second order terms, (2.9)

$$(V_p^{(m)})^2 + (V_q^{(m)})^2 + 2V_p^{(m)} \delta V_p + 2V_q^{(m)} \delta V_q = (V_p^{(m+1)})^2 + (V_q^{(m+1)})^2$$

Equations (2.5) and (2.9) form a pair of simultaneous equations which can be solved for δV_p and δV_q as follows:

Firstly rewriting equations (2.5) and 2.9) as:

$$P = A \delta V_p + B \delta V_q \qquad (2.5.1)$$

$$\text{modV} = V_p \delta V_p + V_q \delta V_q \qquad (2.9.1)$$

where $\text{modV} = 0.5(((V_p^{(m+1)})^2 + (V_q^{(m+1)})^2) - ((V_p^{(m)})^2 + (V_q^{(m)})^2))$

Hence $$\delta V_p = (V_q \delta P - B \text{ modV}) / \Delta \qquad (2.10)$$

$$\delta V_q = (A \delta P - V_p \text{ modV}) / \Delta \qquad (2.11)$$

where $\Delta = A V_q - B V_p$

The term modV will tend to zero as the iteration proceeds, but it may not be zero in the early iterations. For example, if the initial value of V_k is

$$|V_{ks}| + j0, \quad V_{kp}^{(m)} = |V_{ks}| \quad \text{and} \quad V_q^{(m)} = 0$$

therefore, $\text{modV} = (|V_{ks}|^2 - |V_{ks}|^2)$

and $dV_p = (V_q^{(m)} \delta P) / \Delta = 0$

$dV_q = (A \delta P) / \Delta$

the first, improved value of V_k is, therefore,

$$V_k = |V_{ks}| + j\delta V_q$$

for the second iteration $|V_k| \neq |V_{ks}|$ hence modV $\neq 0$.

The subroutine which solves these equations is called PSA7, a detailed flow chart of which is given in Section 6.9.

2.3 Three Phase Short Circuit Analysis

Network symmetry is maintained under three phase short circuit conditions and, as for the balanced load flow, this analysis requires only the positive sequence network.

The short circuit study is treated as a special case of load flow, the voltage at the faulted node being set to zero. The advantage of this approach is that it automatically includes the prefault system conditions.

2.3.1 Short Circuit Analysis - Convergence

This analysis has been carried out on the A.E.P. 14 Bus System,[14] for example, and although an unaccelerated convergence required 213 iterations, (voltage tolerance 1.0×10^{-8}), convergence has always been achieved. This is discussed more fully in 2.4.1.

2.4 Asymmetric Fault Analysis

The asymmetric fault analysis is based on the method of symmetrical components. The procedure for obtaining a solution to the equations of the faulted system is as follows; the positive sequence nodal admittance matrix is modified to form the negative sequence nodal admittance matrix. For static electrical plant the positive and negative sequence impedances are equal, therefore, the only modifications required are those relating to generators and motors.

The zero sequence nodal admittance matrix is formed in a similar manner. However, more operations are required in transforming the positive sequence nodal admittance matrix into the zero sequence nodal admittance matrix because not only are some of the branch values different, but the zero sequence equivalent circuits of the transformers will have different forms depending on the winding connections.

The condition of the faulted system is computed in several stages. Firstly the negative sequence network is analysed, using the following method. A voltage of 1.0 p.u. is applied at the fault position and a load flow analysis is performed. The values of all nodal voltages are recorded. Also, the impedance of this network, as seen from the fault point is evaluated;

$Z_n = 1.0 / I_n$.

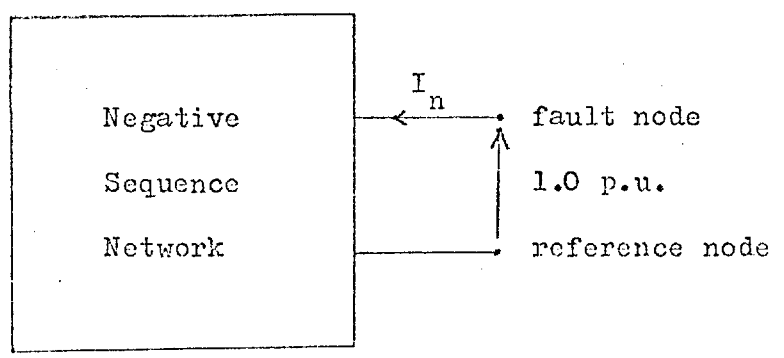


Fig. 2.2.

Secondly, this procedure is repeated, if the fault conditions require it, for the zero sequence network. Again, the nodal voltages are recorded, and the network impedance as seen from the fault position is evaluated.

The negative and zero sequence impedances are combined, appropriately, with any fault impedance andthe resulting impedance value is used to modify the positive sequence network, so that an equivalent circuit for the faulted power system is obtained.

For example, a single line to earth fault requires that the faulted node of the positive sequence network is shunted by $Z_n + Z_o + 3Z_f$, where Z_f is the fault impedance.

A new analysis is then made on the modified positive sequence network, the values of the nodal voltages being recorded. The previously recorded values of nodal voltage, for the negative and zero sequence networks, are now scaled by a factor which is determined from the above analysis.

For example, the arrangement of a modified positive sequence network for a single line to earth fault is as shown by Fig. 2.3.

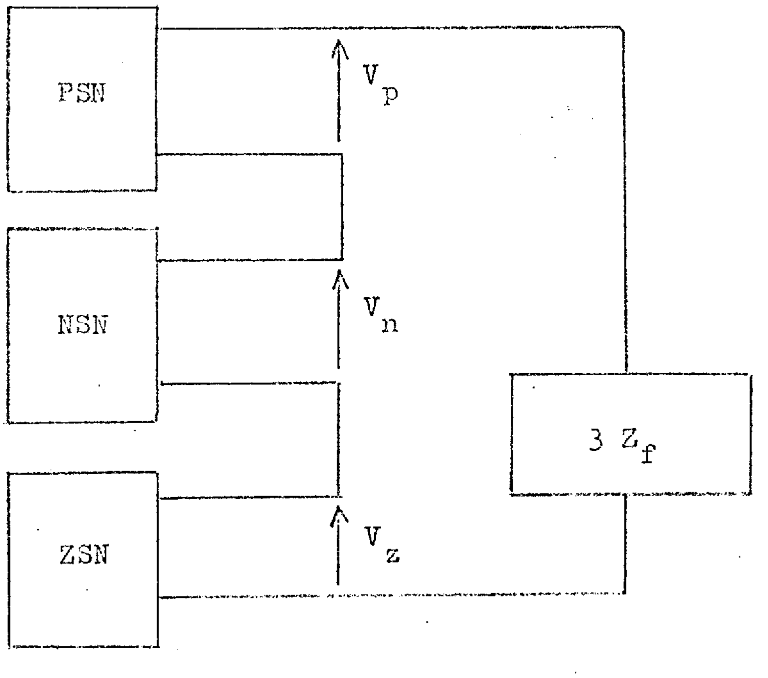


Fig. 2.3.

The originally recorded values of voltage for the negative and zero sequence networks were due to an excitation voltage of 1.0 p.u. From Fig. 2.3 it is seen that the actual value of excitation voltage for the negative sequence network is less than 1.0 p.u. and, as the networks are linear, all voltages in the negative sequence network are scaled by a factor V_n. This procedure is repeated for the zero sequence voltages, with a scale factor V_z.

The sequence currents are computed by multiplying the branch sequence admittances by the corresponding branch sequence voltages. The actual, unbalanced three phase system voltages and currents are then computed from the sequence values using the standard technique that was originally developed by Fortescue (9).

2.4.1 Acceleration and Convergence

Although convergence to solutions with a voltage tolerance of 10^{-8} has always been achieved using the Newton Raphson method, the number of iterations required has sometimes been quite large, for example 213 iterations for the A.E.P. 14 bus system. Since the original work with this program was carried out using a small and relatively slow computer (I.B.M. 1130) the computing times were of the order of 20 minutes, which is too long to be acceptable. In order to reduce the number of iterations, and hence the computing time required, the solution was accelerated by use of the Epsilon Algorithm(27). This algorithm is very successful when applied to the type of sequence obtained from a series of successive voltage iterates, and reduced the number of iterations required for the A.E.P. 14 Bus System from 213 to 56. The computing time required (using an I.C.L. 1905 computer) is approximately 10 minutes for the unaccelerated 213 iterations and 2.5 minutes for 56 iterations.

2.4.1.1 The Epsilon Algorithm

The Epsilon Algorithm can be defined by the following relationship:

$$P_{u,v+\frac{1}{2}} = P_{u,v-\frac{1}{2}} + \frac{1}{P_{u+\frac{1}{2},v} - P_{u-\frac{1}{2},v}}$$

The relationship between the first few values of the given sequence and the algorithm is illustrated by the following table:

u \ v	0	$\frac{1}{2}$	1	$\frac{3}{2}$	2
0	$P_{0,0}$				
$\frac{1}{2}$		$P_{\frac{1}{2},\frac{1}{2}}$			
1	$P_{1,0}$		$P_{1,1}$		
$\frac{3}{2}$		$P_{\frac{3}{2},\frac{1}{2}}$		$P_{\frac{3}{2},\frac{3}{2}}$	
2	$P_{2,0}$		$P_{2,1}$		$P_{2,2}$
$\frac{5}{2}$		$P_{\frac{5}{2},\frac{1}{2}}$		$P_{\frac{5}{2},\frac{3}{2}}$	
3	$P_{3,0}$		$P_{3,1}$		$P_{3,2}$

Where $P_{0,0}$, $P_{1,0}$, $P_{2,0}$ etc. are the original sequence values, $P_{1,1}$, $P_{2,1}$, $P_{3,1}$ etc. is the first sequence of accelerated values and $P_{2,2}$, $P_{3,2}$, $P_{4,2}$ etc. is the second sequence of accelerated values.

2.4.1.2 <u>An Illustrative Example</u>

Consider the following 2 node system:

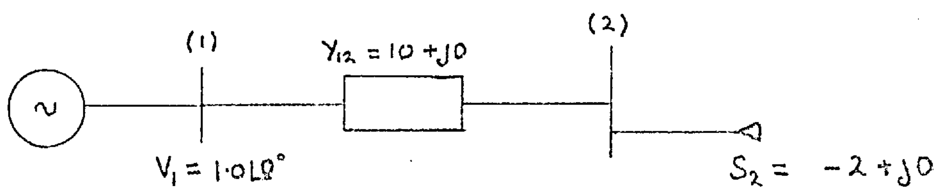


Fig. 2.5

The iterative equation for the voltage at node 2 is

$$V_2^{(n+1)} = \frac{1}{10}\left[\left[\frac{-S}{V_2^{(n)}}\right]^* + 10 \times 1.0\right]$$

$$V_2^{(n+1)} = \frac{-0.2}{V_2^{(n)}} + 1.0$$

and the following list of function (V_2) values has been derived from this equation, using a starting value of 1.0 for V_2 .

Iterate	Function Value
0	1.0000000000
1	0.8000000000
2	0.7500000000
3	0.7333333333
4	0.7272727272
5	0.7250000000
.	
.	
.	
25	0.72360679776
26	0.72360679775
27	0.72360679775

The first sequence of accelerated values is obtained by adding to the original function value the reciprocal of the differences of the reciprocal of the differences. For example, taking the first three function values:

Function Value	Δ_1	$1/\Delta_1$	Δ_2	$1/\Delta_2$	Accelerated Value
1.0					
	-0.2	-5.0			
0.8			-15.0	-0.066666	0.7333333
	-0.05	-20.0			
0.75					

where the accelerated value $P_{1,1}$ is given by

$$P_{1,1} = 0.8 + 1/\Delta_2$$

The sequence of accelerated values is extended by using iterates 2,3 and 4; 3,4 and 5, etc. Iterates 8, 9 and 10 yield a value which is in error by 1 in the eigth decimal place, 0.7236068111.

The algorithm can also be applied to the sequence of accelerated values, to form the function values $P_{2,2}$, $P_{3,2}$, $P_{4,2}$ etc. The sixth value in this sequence, which is computed from the sixth, seventh and eighth values of the first accelerated sequence is 0.72360679771 i.e. correct to ten decimal places.

The 'Error Analysis' of this algorithm is difficult. However, it can be seen that when the function values agree to, say, six decimal places, the differences are of the order 10^{-6} and hence the reciprocal of the differences become relatively large, of the order 10^{6}. Thus the evaluation of an accelerated value in this instance requires a computation involving numbers having their most significant figures displaced by 10^{12} on the decimal scale. Experience has shown that normal precision working (11 significant figures) is insufficient for the analysis of practical problems if the second sequence of improved values is used to accelerate the solution; convergence for the A.E.P. 14 Bus System, when attainable, taking over 400 iterations. However, as stated previously, the first sequence of improved values can be computed using normal precision, and used in practical analysis to advantage.

2.5 Equivalent Circuits [31]

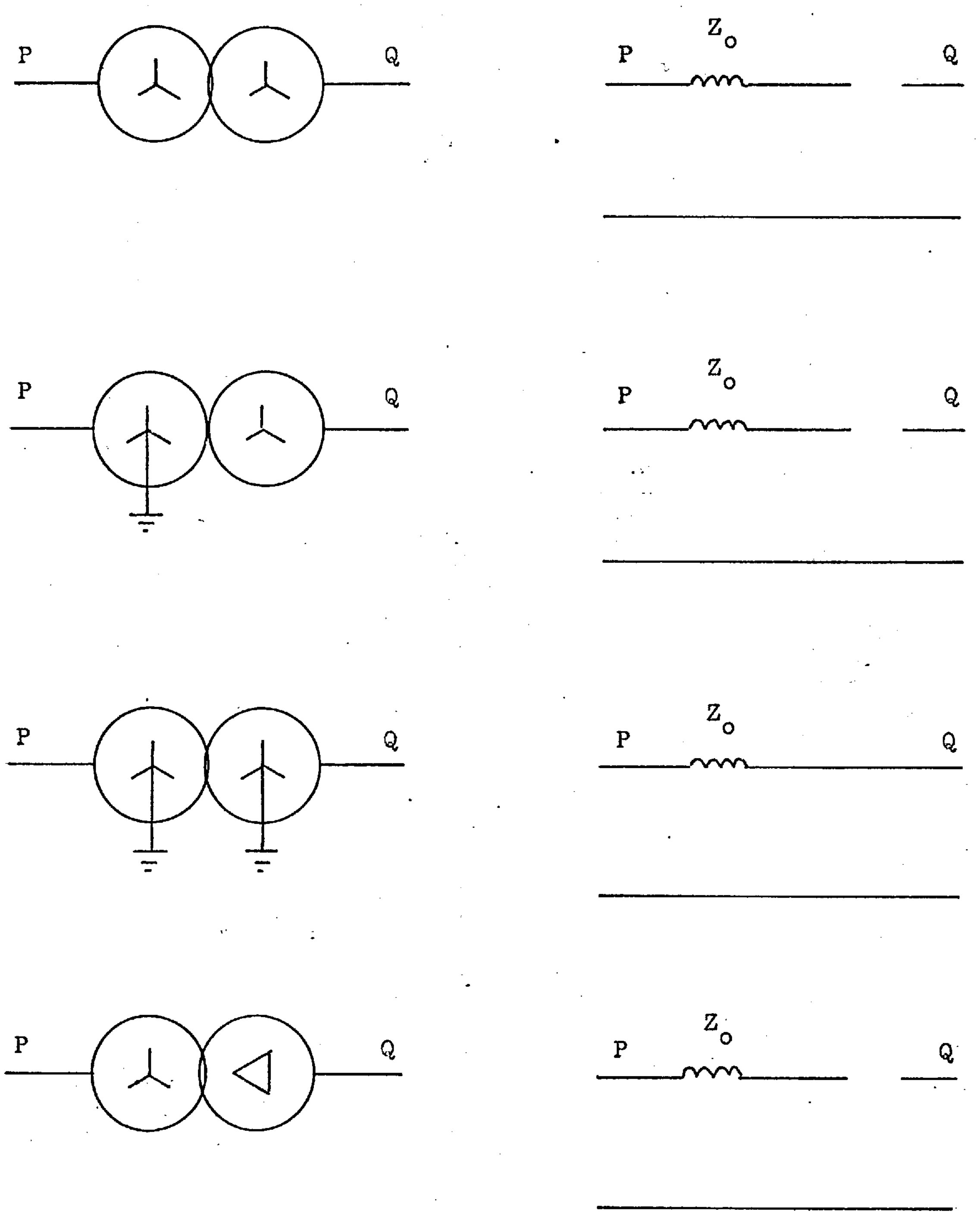
The model of the system network is dependent upon the equivalent circuits used to represent the various items of plant. This study utilises simplified lumped parameter equivalent circuits, an approach which is justified since the error introduced is much less than the tolerance to which the system data is usually known.

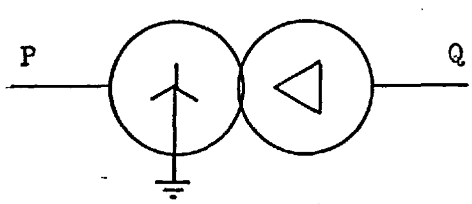
2.5.1 Two Winding Transformer Equivalent Circuits

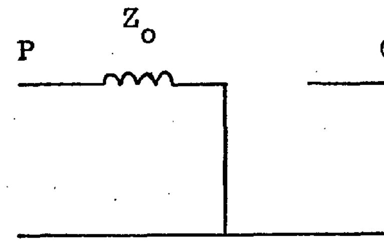
The following equivalent circuits have been derived by neglecting the magnetising current circuit and the magnetic path provided by the transformer tank. The later assumption is justified since the zero sequence impedance of such a path is usually about twenty times that of the leakage impedance. For the positive and negative sequence circuits the effect of this path should be negligable.

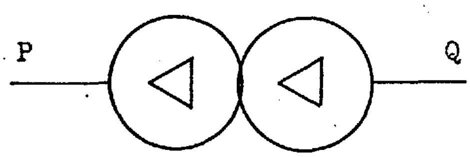
2.5.1.1 Zero Sequence Equivalent Circuits

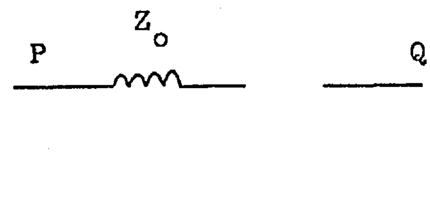
Connection Diagram | Equivalent Circuit





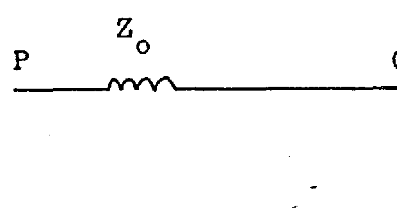






2.5.1.2 <u>The Positive Sequence Equivalent Circuit</u>

The positive sequence equivalent circuits of the above transformers are identical:



2.5.1.3 <u>The Negative Sequence Equivalent Circuit</u>

The negative sequence equivalent circuit is identical with the positive sequence circuit.

2.5.2 <u>Three Winding Transformers</u>

Three winding transformers are represented by the conventional 'T' equivalent circuit.

2.5.2.1 <u>The Positive and Negative Sequence Equivalent Circuits</u>

These equivalent circuits are identical and take the following form, for any mode of winding connections.

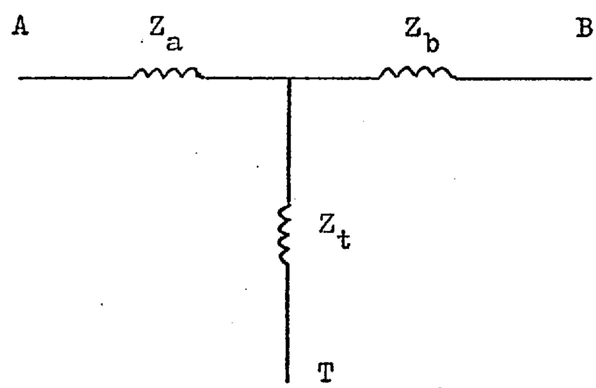


Fig. 2.4

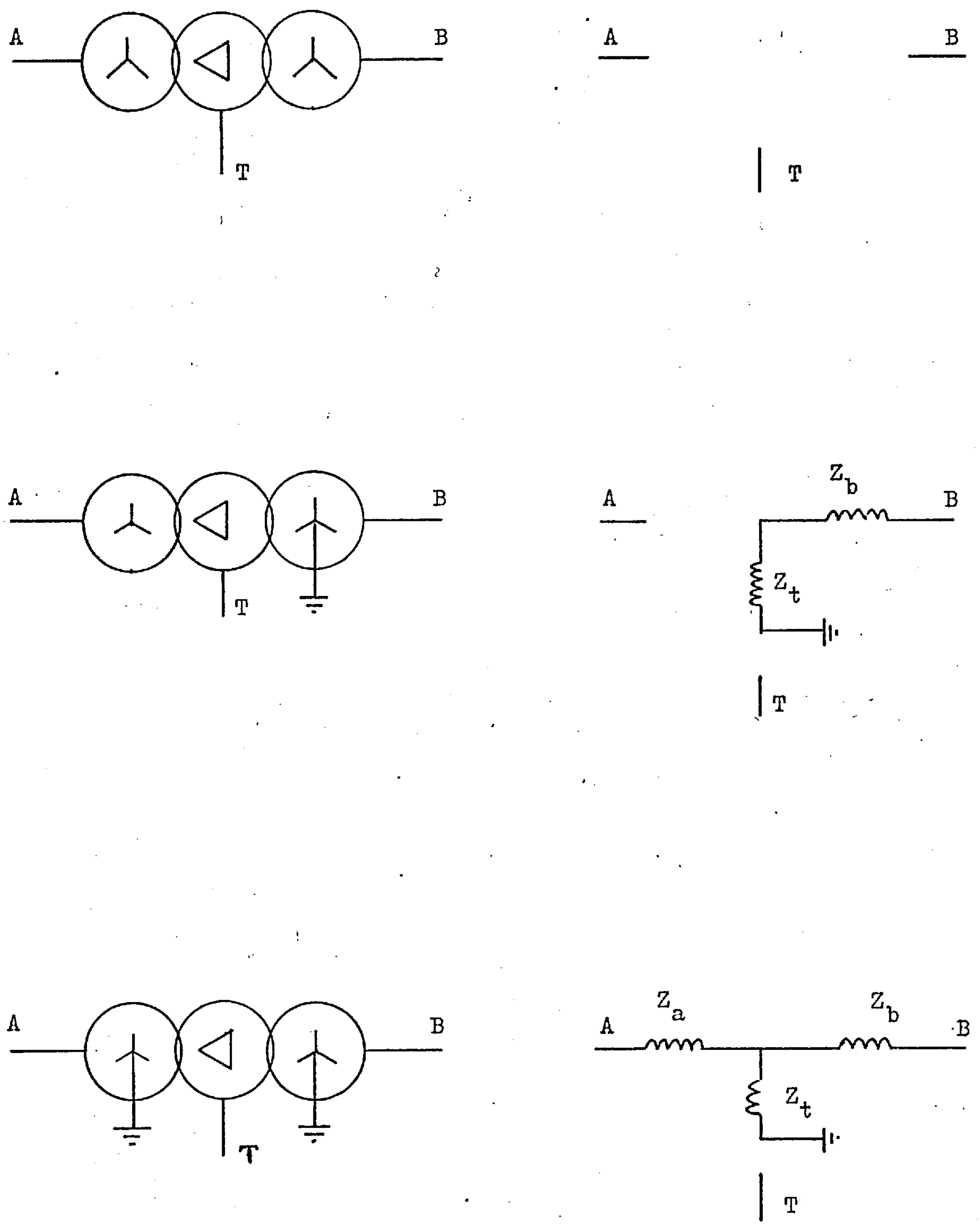
Where $Z_a = 0.5 \ (Z_{ps} + Z_{pt} - Z_{st})$

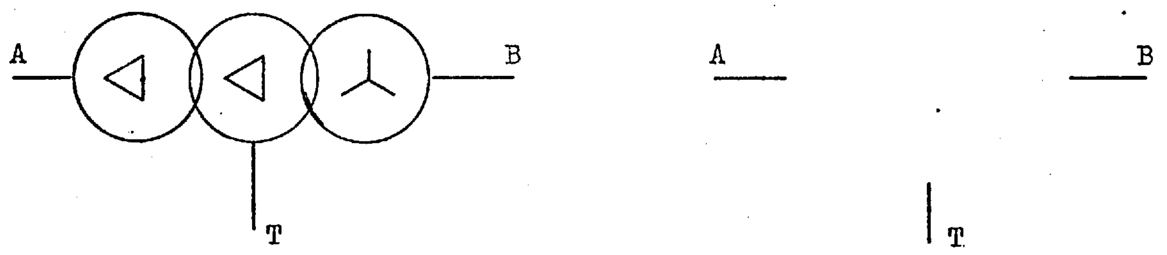
$Z_b = 0.5 \ (Z_{ps} + Z_{st} - Z_{pt})$

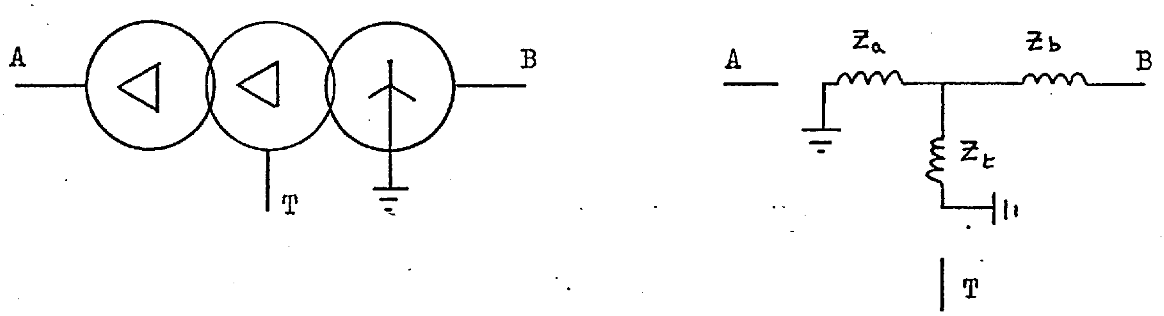
$Z_c = 0.5 \ (Z_{pt} + Z_{st} - Z_{ps})$

and Z_{ps} is the leakage impedance measured at the terminals of the primary winding, with the secondary winding short - circuited and the tertiary winding open - circuited. Z_{pt} is the leakage impedance measured at the terminals of the primary winding, with the tertiary winding short -circuited and the secondary winding open - circuited. Z_{st} is the leakage impedance measured at the terminals of the secondary winding, with the tertiary winding short-circuited and the primary winding open-circuited.

2.5.2.2 Zero Sequence Equivalent Circuits




A
B
T
A
B
T


A
B
T
A
Z_a
Z_b
B
Z_t
T

2.5.3 Motors and Generators

One of the objectives of this work has been to obtain an approximation for the variation of the effective reactance of induction motors and synchronous motors or generators, by carrying out a minimum number of load flows. It is suggested that three values of reactance, initially corresponding to X''_d, X'_d and X_s, would be sufficient for this purpose. The equivalent circuit used to model the electrical behaviour of these machines is, therefore, a constant voltage source acting behind a time varying reactance as shown by Fig. 2.5. The use of this equivalent circuit has been justified by M.G. Say [10] in his standard text on alternating current machines.

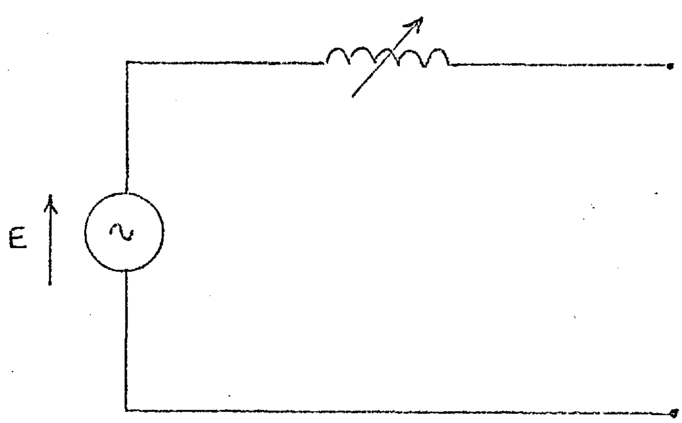


Fig. 2.5

The value of the reactance X is usually determined from measurements made on an actual machine, or obtained from tables of typical values [6, 13].

2.5.3.1 Typical Constants for Synchronous Generators

Reactance Values (P.U.)

	X''_d	X'_d	X_s	X_2	X_o
A 4-Pole turbine generators	0.14	0.23	1.2	0.14	0.08
B 2-Pole turbine generators	0.09	0.15	1.2	0.09	0.03
C Salient pole generators (with dampers)	0.2	0.3	1.25	0.2	0.18

Time Constants (seconds)

	T''	T'	T_a
A	0.035	0.6	0.13
B	0.35	1.0	0.2
C	0.035	1.5	0.15

2.5.3.2 Typical Constants for Induction Motors

Reactance values in per unit, and time constants in seconds.

Machine Rating h.p.	X"	X'	T"	T'
22 000	0.182	0.216	0.0076	0.077
6040	0.242	0.294	0.0176	0.114
2500	0.283	0.378	0.025	0.141

2.5.3.3 Induction Motors

These parameters can be used to generate the instantaneous values of current; for example Kalsi (6) gives the following expression for an induction motor which is subject to an indirect short-circuit;

$$i = E_{max}\left(\frac{1}{X''_d} - \frac{1}{X'_d}\right)\exp(-t/T'')$$
$$+ \left(\frac{1}{X'_d}\right)\exp(-t/T')\sin(wt + \lambda)$$
$$- \left(\frac{1}{X''_d}\right)\sin(\lambda)\exp(-t/T_a) \qquad (2.12)$$

If the RMS value for each period of this current is defined as

$$I = \left(\frac{1}{T}\int_{KT}^{(K+1)T} f(t)^2 dt\right)^{\frac{1}{2}} \qquad k\ 0,1,2,3,\ \ldots \qquad (2.13)$$

the values obtained form, approximately, an exponentially decreasing sequence as shown in Fig. 2.6.

Typical data [12] used with equation(2.12) is:-

X''_d = 0.179

X'_d = 0.25

T''_d = 0.014

T'_d = 0.132

T_a = 0.033

The variation of I_{RMS} can be obtained by using the equivalent circuit of Fig. 2.5 with an appropriately varying reactance. For the RMS current function shown in Fig. 2.6, the corresponding reactance function is shown in Fig. 2.7. This reactance function given by $X(t) = \frac{E}{I(t)}$ can be approximated, over the range indicated, by the function

$$X(t) = A + B.t + C.(t)^{\frac{1}{2}} \tag{2.14}$$

(see chapter 3), where in this case

$A = 0.09862$

$B = 4.02570$

$C = -0.02762$

Values from this function are also indicated in Fig.2.7.

2.5.3.4 Synchronous Alternators

Adkins[11] gives the following approximate equation for the fault current of a synchronous alternator with a line to line fault:

$$i_B = \sqrt{3}\, E\left[\frac{1}{X_d + X_2} + \left(\frac{1}{X'_d + X_2} - \frac{1}{X_d + X_2}\right)\exp(-t/T') + \left(\frac{1}{X''_d + X_2} - \frac{1}{X'_d + X_2}\right)\exp(-t/T'') \quad \cos(wt + \lambda)\right] \tag{2.15}$$

This equation has been derived by neglecting the harmonic current components. It should be noted that when the RMS response contains a d.c. component the reactance function which is used to reproduce this current will not equal X" and X' at times $t = 0$ and $t = T''$ respectively. The positive sequence reactance function corresponding with the RMS values of 2.15 is shown in Fig. 2.9 with values from the approximating function

$$X(t) = 0.1333 + 6.4719t - 0.6494(t)^{\frac{1}{2}} \tag{2.16}$$

The positive sequence current for a line to line fault with $X_2 = 0.1$, obtained by using 2.15, is shown in Fig. 2.8.

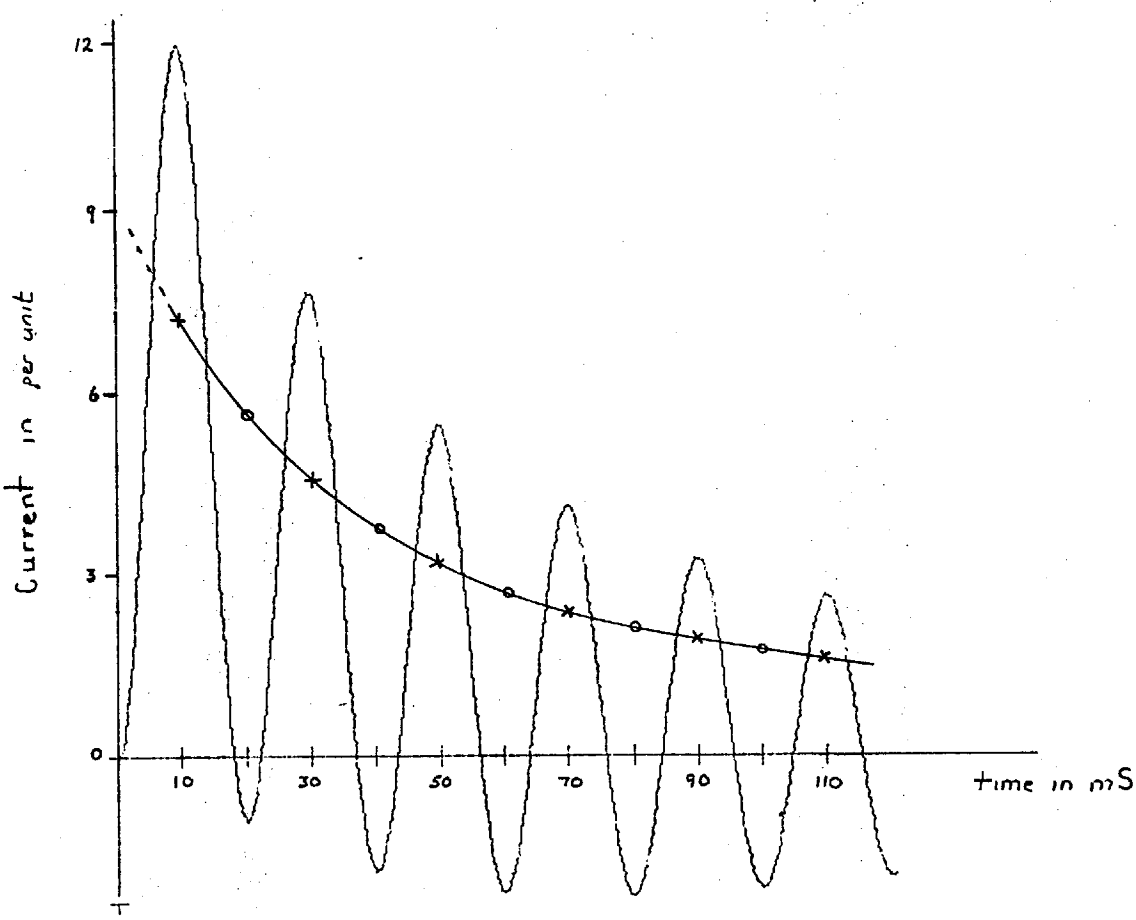
Data [11] used with equation 2.15

X''_d	= 0.1	T''_d	= 0.035
X'_d	= 0.15	T'_d	= 0.06
X_d	= 1.2	T_a	= 0.13
X_2	= 0.1	λ	= $-\pi/2$

2.6 The Operation of the Proposed Method

The author proposes that the effective R.M.S. current function be approximated by equation 2.16, as described in section 3.8, so that when used with the relay characteristics, the relay performance can be evaluated, see section 3.11.

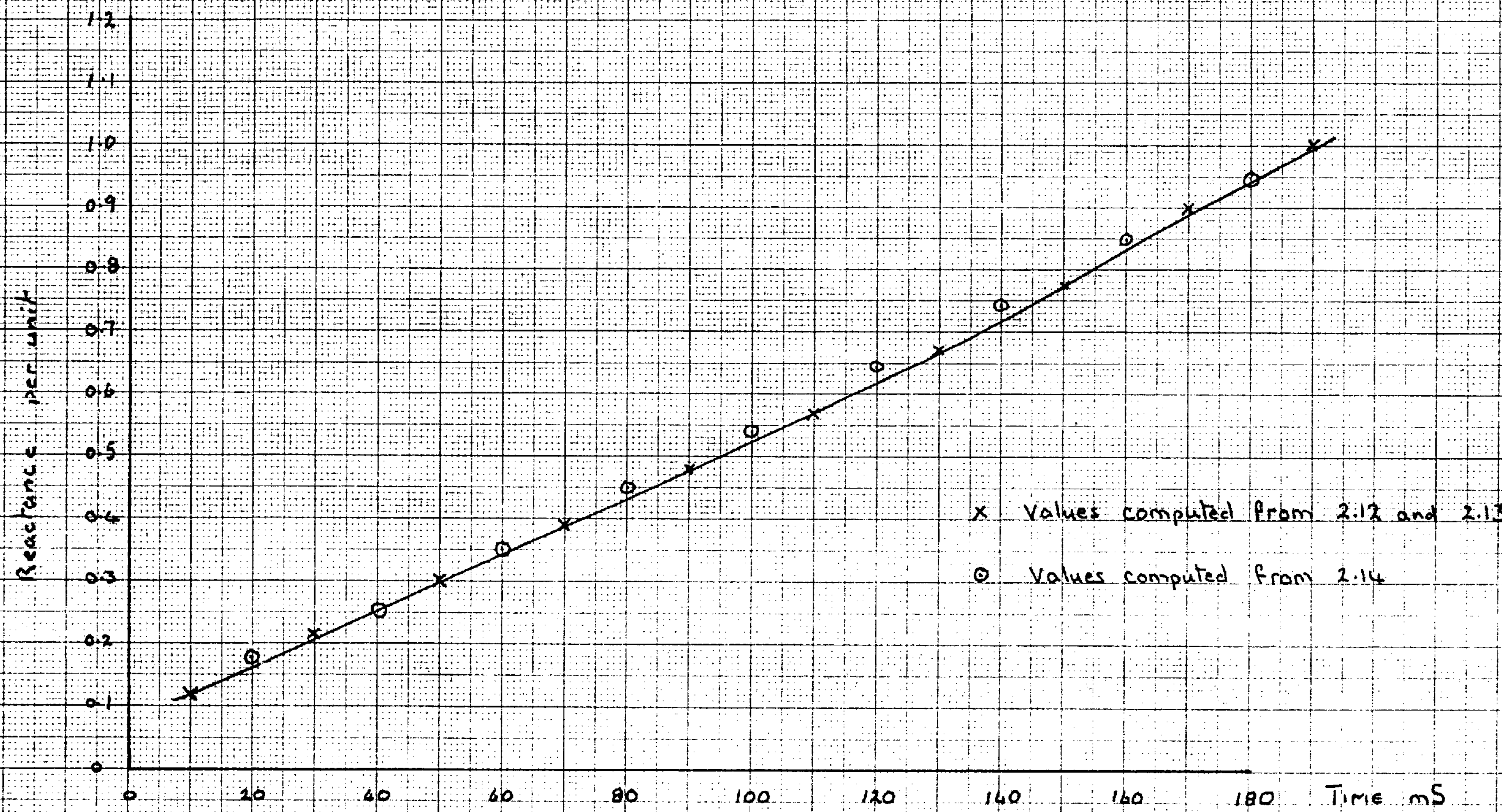
Instantaneous and RMS Current for an Indirect Short Circuit (3 phase to earth) at the Induction Motor Terminals



X R.M.S values computed from 2.13 and 2.12

O R.M.S values computed using 2.14

Fig. 2.6

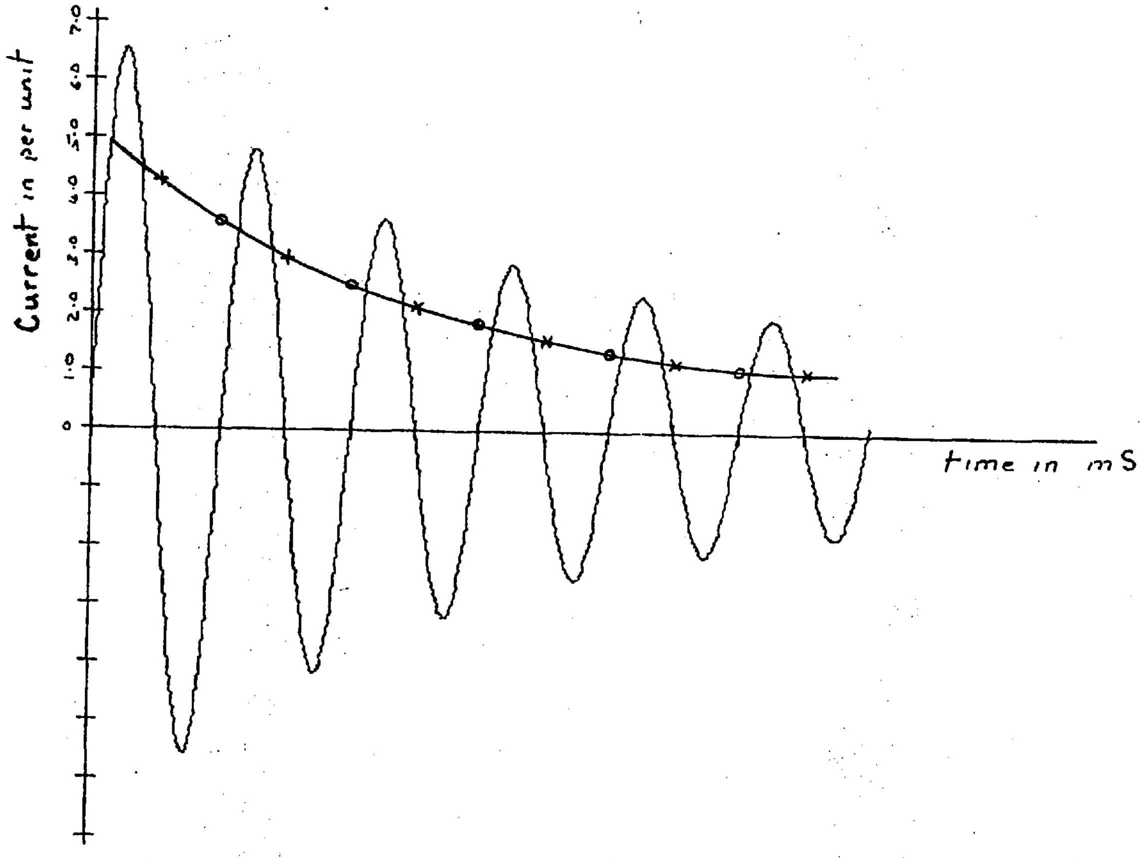


THE DERIVED REACTANCE FUNCTION FOR THE INDUCTION MOTOR

Fig. 2.7

Instantaneous and RMS Fault Current for a

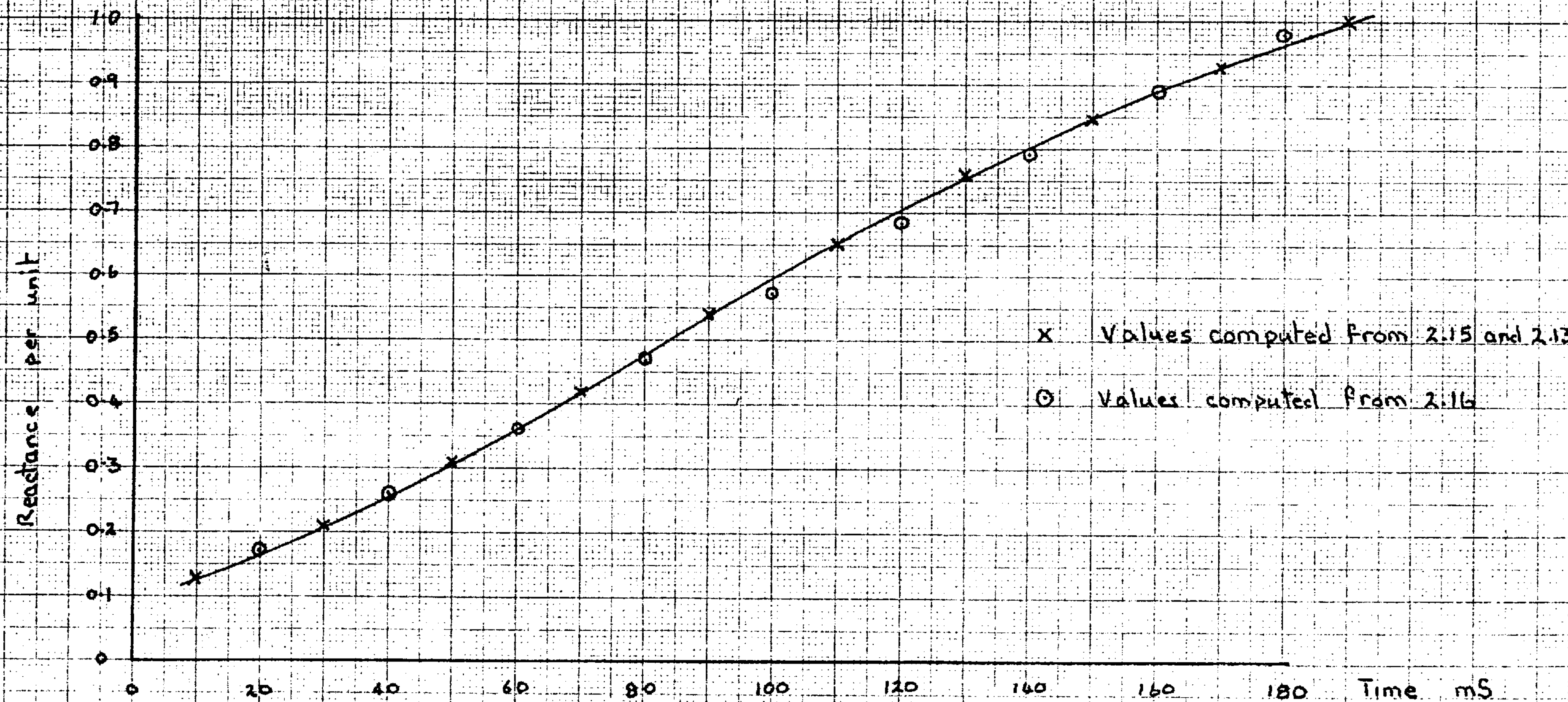
Line to Line Fault on a Synchronous Alternator



X RMS values computed from 2.15 and 2.13

O RMS values computed from 2.16

Fig. 2.8



THE DERIVED SYNCHRONOUS ALTERNATOR REACTANCE FUNCTION.

Fig 2.9

2.7 14 Bus Test System

This test system is based on the A.E.P. 14 bus system for which only the positive sequence system data is published. The zero sequence data was determined by using a table of typical values and comparing the values of the X/R ratios for the lines of the test system with those of standard lines.

The System Diagram is shown by Fig. 2.10 and the system data is shown by the listing of the data cards, 2.7.1.

The applied protective system is shown by Fig.2.11, and is comprised of 10 I.D.M.T. relays, all using the TJX Overcurrent/Earth Fault 1 Amp. relay characteristic which has a definite minimum time of 3 seconds. This characteristic is listed in section 2.13.1. The time - multiplier settings are shown on the diagram, Fig. 2.11 and the plug bridges are set to 100 per cent.

The results of the analysis show that the fault is isolated from the system by the operation of relays.

14 Bus Test System - Line Diagram

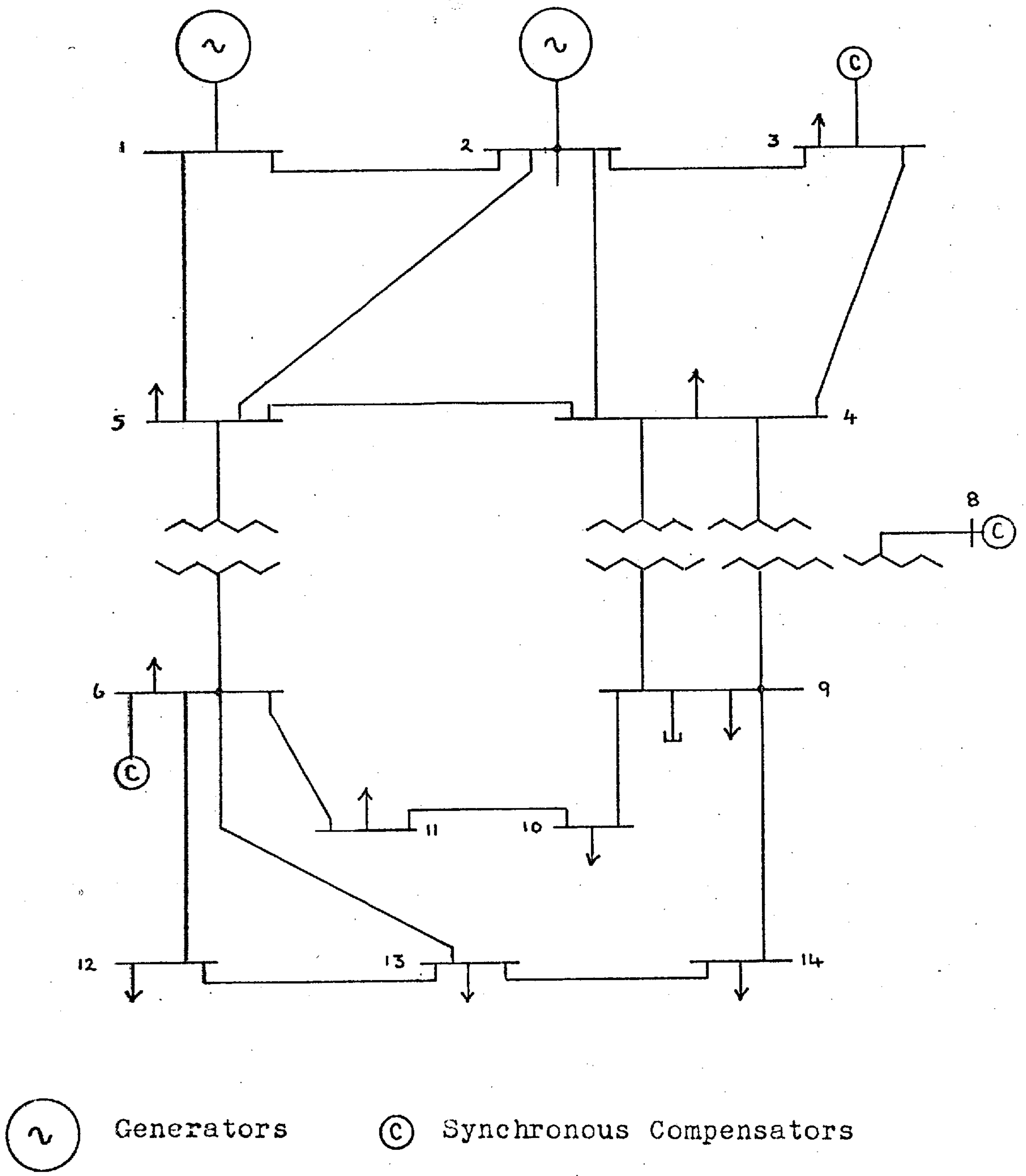
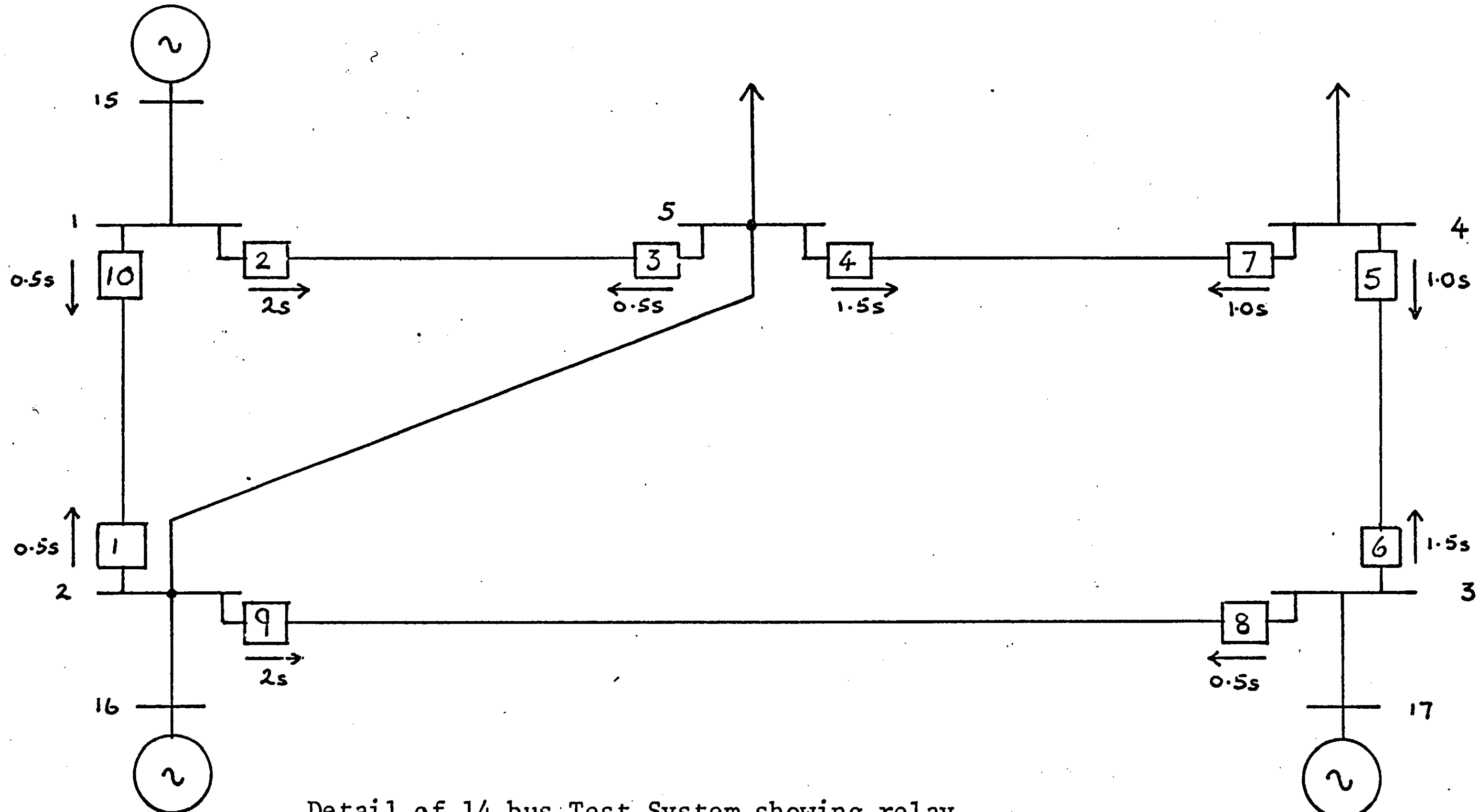


Fig. 2.10



Detail of 14 bus Test System showing relay locations and time settings.

Fig. 2.11

Reference Section

```
       13/06/75                                   12/17/29
*JOB NUMBER                                        9
*DATA
*ANALYSIS      PROTECTION                          L/L/E        20     3.0
*PRINT DATA
*END
    /*JOB NAME          14 BUS SYSTEM EXTENDED TO INCLUDE M/C REACTANCES
```

Job control cards 5.6

```
    FOR THIS STUDY

    THE VOLTAGE' TOLERANCE IS         0.100000E-05

    THE TRACE LEVEL IS    0

    A SHORT CIRCUIT STUDY WILL BE MADE AT      0  BUSBARS
```

5.7. a

P	Q	MODV**2	MOD AND ANGLE OF Y		VP	VQ	ND7	MPS	J	ND	NDZ
0.0000	0.0000	0.0000	0.0510	90.0000	0.8990	0.0000	1	0	1	1	1
0.0000	0.0000	0.0000	0.2130	-9.2330	0.8990	0.0000	1	0	2	1	1
-0.9420	0.0000	0.0000	0.1470	-90.0000	0.8990	0.0000	1	0	3	1	1
-0.4780	0.0390	0.0000	0.0374	90.0000	0.8990	0.0000	1	0	4	1	1
-0.0760	-0.0160	0.0000	0.0340	90.0000	0.8990	0.0000	1	0	5	1	1
-0.1120	0.0000	0.0000	0.0655	-90.0000	0.8990	0.0000	1	0	6	1	1
0.0000	0.0000	0.0000	0.0000	0.0000	0.8990	0.0000	1	0	7	1	1
0.0000	0.0000	0.0000	0.0000	0.0000	0.8990	0.0000	1	0	8	1	1
-0.2950	-0.1660	0.0000	0.1900	90.0000	0.8990	0.0000	1	0	9	1	1
-0.0900	-0.0580	0.0000	0.0000	0.0000	0.8990	0.0000	1	0	10	1	1
-0.0350	-0.0180	0.0000	0.0000	0.0000	0.8990	0.0000	1	0	11	1	1
-0.0610	-0.0160	0.0000	0.0000	0.0000	0.8990	0.0000	1	0	12	1	1
-0.1350	-0.0580	0.0000	0.0000	0.0000	0.8990	0.0000	1	0	13	1	1
-0.1490	-0.0500	0.0000	0.0000	0.0000	0.8990	0.0000	1	0	14	1	1
0.0000	0.0000	0.0000	0.0000	0.0000	0.7243	2.6335	0	0	15	-1	-1
0.0000	0.0000	0.0000	0.0000	0.0000	1.2283	0.3551	0	0	16	-1	-1
0.0000	0.0000	0.0000	0.0000	0.0000	1.1066	-0.2459	0	0	17	-1	-1
0.0000	0.0000	0.0000	0.0000	0.0000	1.7652	-0.4776	0	0	18	-1	-1
0.0000	0.0000	0.0000	0.0000	0.0000	1.3245	-0.3243	0	0	19	-1	-1
0.0000	0.0000	0.0000	0.0000	0.0000	0.8990	0.0000	1	0	20	1	1

5.7.d

```
26    20    3    5    1    0        BB   N   TF   ICONT   ISWGN   IEMF
```

5.7.c

BRANCH DATA

BRANCH		RESISTANCE	REACTANCE
1	15	0.0000	1.2000
2	16	0.0000	1.2000
3	17	0.0000	1.2000
6	18	0.0000	1.2000
8	19	0.0000	1.2000
1	20	0.0097	0.0296
20	2	0.0097	0.0296
1	5	0.0540	0.2230
2	3	0.0470	0.1980
2	4	0.0581	0.1765
2	5	0.0570	0.1739
3	4	0.0670	0.1710
4	5	0.0133	0.0421
4	7	0.0000	0.2091
4	9	0.0000	0.5562
5	6	0.0000	0.2520
6	11	0.0950	0.1989
6	12	0.1229	0.2558
6	13	0.0661	0.1303
7	8	0.0000	0.1761
7	9	0.0000	0.1100
9	10	0.0318	0.0845
9	14	0.1271	0.2704
10	11	0.0820	0.1921
12	13	0.2209	0.1999
13	14	0.1709	0.3480

5.7.f

T/F	K	M
0.9320	5	6
0.9780	4	7
0.9690	4	9

5.7.g

Balanced Load Flow

NET GENERATIONS OR LOAD AT BUSBARS

	P.U.VOLTAGES		P.U. GENERATION		P.U. LOAD	
BUS	MOD	ANGLE (DEG)	MW	MVAR	MW	MVAR
1	1.0562	359.79			0.0000	0.0000
2	1.0414	354.83			0.0000	0.0000
3	1.0069	347.22			0.9420	0.0000
4	1.0371	349.22			0.4780	-0.0390
5	1.0449	350.64			0.0760	0.0160
6	1.0673	344.65			0.1120	0.0000
7	1.0469	346.02			0.0000	0.0000
8	1.0874	346.06			0.0000	0.0000
9	1.0400	344.36			0.2950	0.1660
10	1.0374	344.13			0.0900	0.0580
11	1.0487	344.27			0.0350	0.0180
12	1.0514	343.81			0.0610	0.0160
13	1.0458	343.77			0.1350	0.0580
14	1.0244	343.08			0.1490	0.0500
15	2.7313	74.62	2.3203	5.5876		
16	1.2786	16.12	0.4029	0.3285		
17	1.1336	347.47	0.0042	0.1197		
18	1.8287	344.86	0.0059	1.1603		
19	1.3636	346.24	0.0040	0.3138		
20	1.0478	357.33			0.0000	0.0000

LINE FLOWS. (ALL VALUES IN P.U.)

LINE	SEND		RECEIVE		CURRENT	
	MW	MVAR	MW	MVAR	MOD	ANGLE
1 - 5	0.7700	-0.0701	0.7410	-0.1896	0.7320	4.99
1 - 15	-2.3203	0.3006	-2.3203	-5.5876	2.2151	187.17
1 - 20	1.5503	-0.1735	1.5291	-0.2381	1.4769	6.18
2 - 3	0.7155	0.0584	0.6932	-0.0357	0.6894	350.16
2 - 4	0.5559	-0.1286	0.5384	-0.1816	0.5479	7.86
2 - 5	0.4115	-0.1393	0.4016	-0.1696	0.4172	13.54
2 - 16	-0.4029	-0.1301	-0.4029	-0.3285	0.4066	156.94
2 - 20	-1.5079	0.3027	-1.5291	0.2381	1.4769	186.18
3 - 4	-0.2446	-0.0784	-0.2490	-0.0895	0.2551	149.44
3 - 17	-0.0042	-0.1063	-0.0042	-0.1197	0.1057	79.50
4 - 5	-0.6314	0.0151	-0.6363	-0.0005	0.6089	170.59
4 - 7	0.2835	-0.0397	0.2835	-0.0560	0.2760	357.19
4 - 9	0.1594	0.0015	0.1594	-0.0121	0.1537	348.70
5 - 6	0.4303	-0.0640	0.4303	-0.1109	0.4163	359.11
6 - 11	0.0698	0.0669	0.0690	0.0652	0.0906	300.88
6 - 12	0.0783	0.0292	0.0775	0.0276	0.0783	324.23
6 - 13	0.1762	0.0883	0.1739	0.0838	0.1846	318.04
6 - 18	-0.0059	-0.6772	-0.0059	-1.1603	0.6345	75.16
7 - 8	-0.0040	-0.2409	-0.0040	-0.2502	0.2301	76.97
7 - 9	0.2874	0.0696	0.2874	0.0609	0.2825	332.40
8 - 19	-0.0040	-0.2502	-0.0040	-0.3138	0.2301	76.97
9 - 10	0.0563	0.0116	0.0562	0.0114	0.0553	332.71
9 - 14	0.0954	0.0164	0.0943	0.0141	0.0931	334.60
10 - 11	-0.0338	-0.0466	-0.0340	-0.0472	0.0555	110.02
12 - 13	0.0165	0.0116	0.0165	0.0115	0.0192	308.79
13 - 14	0.0554	0.0373	0.0547	0.0359	0.0639	309.78

3 20 0.0000 0.0000FAULT TYPE FAULT BUS FAULT IMPEDANCE 5.7.i
5 THE NUMBER OF NEGATIVE SEQUENCE MODS 5.7.j

1	15	0.000000E 00	-0.500000E 01	NEGATIVE SEQUENCE MODS	5.7. k
2	16	0.000000E 00	-0.500000E 01	NEGATIVE SEQUENCE MODS	
3	17	0.000000E 00	-0.500000E 01	NEGATIVE SEQUENCE MODS	
6	18	0.000000E 00	-0.500000E 01	NEGATIVE SEQUENCE MODS	
8	19	0.000000E 00	-0.500000E 01	NEGATIVE SEQUENCE MODS	

46 ITERATIONS FOR NEGATIVE NETWORK

THE NEGATIVE SEQUENCE IMPEDANCE IS 0.0107 + J(0.0785) (As seen from the fault point)

Zero Sequence Network Data

K M TFCON ZR ZX XZS XZT MN NT	1	2	2	0.4700	0.1480	0.0000	0.0000	0	0	5.7.m
K M TFCON ZR ZX XZS XZT MN NT	1	5	2	0.1350	0.5580	0.0000	0.0000	0	0	
K M TFCON ZR ZX XZS XZT MN NT	2	3	2	0.1130	0.4950	0.0000	0.0000	0	0	
K M TFCON ZR ZX XZS XZT MN NT	2	4	2	0.1400	0.4340	0.0000	0.0000	0	0	
K M TFCON ZR ZX XZS XZT MN NT	2	5	2	0.1420	0.4350	0.0000	0.0000	0	0	
K M TFCON ZR ZX XZS XZT MN NT	3	4	2	0.1380	0.4280	0.0000	0.0000	0	0	
K M TFCON ZR ZX XZS XZT MN NT	4	5	2	0.0330	0.1050	0.0000	0.0000	0	0	
K M TFCON ZR ZX XZS XZT MN NT	6	11	2	0.1960	0.5000	0.0000	0.0000	0	0	
K M TFCON ZR ZX XZS XZT MN NT	6	12	2	0.1360	0.3250	0.0000	0.0000	0	0	
K M TFCON ZR ZX XZS XZT MN NT	9	10	2	0.0660	0.2110	0.0000	0.0000	0	0	
K M TFCON ZR ZX XZS XZT MN NT	9	14	2	0.2620	0.6750	0.0000	0.0000	0	0	
K M TFCON ZR ZX XZS XZT MN NT	10	11	2	0.1690	0.4800	0.0000	0.0000	0	0	
K M TFCON ZR ZX XZS XZT MN NT	12	13	2	0.2210	0.2000	0.0000	0.0000	0	0	

K	M	TFCON	ZR	ZX	XZS	XZT	MN	NT	12	13	2	0.2210	0.2000	0.0000	0.0000	0	0
K	M	TFCON	ZR	ZX	XZS	XZT	MN	NT	13	14	2	0.3520	0.8700	0.0000	0.0000	0	0
K	M	TFCON	ZR	ZX	XZS	XZT	MN	NT	1	15	2	0.0000	0.0800	0.0000	0.0000	0	0
K	M	TFCON	ZR	ZX	XZS	XZT	MN	NT	2	16	2	0.0000	0.0800	0.0000	0.0000	0	0
K	M	TFCON	ZR	ZX	XZS	XZT	MN	NT	3	17	2	0.0000	0.0800	0.0000	0.0000	0	0
K	M	TFCON	ZR	ZX	XZS	XZT	MN	NT	6	18	2	0.0000	0.0800	0.0000	0.0000	0	0
K	M	TFCON	ZR	ZX	XZS	XZT	MN	NT	8	19	2	0.0000	0.0800	0.0000	0.0000	0	0

45 ITERATIONS FOR THE ZERO SEQUENCE NETWORK

THE ZERO SEQUENCE IMPEDANCE IS 0.0099 + J(0.0506) (As seen from fault point)

5.7.n

POSITIVE SEQUENCE NETWORK MODIFICATIONS

REACTANCE AND TIME VALUES	1	15	0.2000	0.3300	1.2000	0.0000	0.1050	2.0000
REACTANCE AND TIME VALUES	2	16	0.2000	0.3300	1.2000	0.0000	0.1050	2.0000
REACTANCE AND TIME VALUES	3	17	0.2000	0.3300	1.2000	0.0000	0.1050	2.0000
REACTANCE AND TIME VALUES	6	18	0.2000	0.3300	1.2000	0.0000	0.1050	2.0000
REACTANCE AND TIME VALUES	8	19	0.2000	0.3300	1.2000	0.0000	0.1050	2.0000
REACTANCE AND TIME VALUES	0	0	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Asymmetric Fault Analysis Using X"

LINE CURRENTS

LINE		PHASE A MODULUS	PHASE A ANGLE	PHASE B MODULUS	PHASE B ANGLE	PHASE C MODULUS	PHASE C ANGLE
1	5	2.54397	23.83	2.07377	286.04	2.35219	167.32
		2.32712	1.02776	0.57311	-1.99300	-2.29481	0.51640
1	15	7.31126	199.55	12.25056	39.17	13.76108	292.05
		-6.88976	-2.44658	9.49760	7.73769	5.16646	-12.75442
1	20	4.77535	16.50	11.64740	209.67	12.58096	103.34
		4.57880	1.35592	-10.12085	-5.76459	-2.90199	12.24169
2	3	2.04079	20.94	1.59324	313.83	1.93850	189.22
		1.90602	0.72933	1.10323	-1.14948	-1.91347	-0.31053
2	4	1.54368	35.71	1.41483	320.38	1.61626	198.20
		1.25343	0.90104	1.08988	-0.90217	-1.53537	-0.50490
2	5	1.22684	41.21	1.26143	326.36	1.39134	203.63
		0.92299	0.80822	1.05012	-0.69890	-1.27464	-0.55777
2	16	2.53572	2.90	6.70888	344.58	5.59429	252.43
		2.53247	0.12848	6.46720	-1.78447	-1.68890	-5.33326
2	20	7.29076	201.22	10.66242	154.48	9.28410	45.87
		-6.79641	-2.63895	-9.62200	4.59395	6.46392	6.66424
3	4	0.62935	195.83	0.20164	113.67	0.68657	358.91
		-0.60550	-0.17162	-0.08094	0.18468	0.68644	-0.01306
3	17	2.09163	26.76	2.23505	333.73	2.46846	200.71
		1.86755	0.94190	2.00414	-0.98936	-2.30898	-0.87289
4	5	1.38715	201.52	0.81530	93.51	1.37452	347.19
		-1.29041	-0.50894	-0.04988	0.81377	1.34029	-0.30484
4	7	1.31331	30.50	1.21751	316.88	1.42956	190.79
		1.13159	0.66655	0.88868	-0.83221	-1.40428	-0.26765

4	9	0.50082	57.27	0.50082	297.27	0.50082	177.27
		0.27078	0.42130	0.22947	-0.44515	-0.50025	0.02385
5	6	2.15625	31.58	2.12199	309.43	2.24331	186.08
		1.83696	1.12916	1.34763	-1.63913	-2.23070	-0.23747
6	11	0.47921	269.85	0.47921	149.85	0.47921	29.85
		-0.00126	-0.47921	-0.41438	0.24069	0.41564	0.23852
6	13	0.35555	290.69	0.35555	170.69	0.35555	50.69
		0.12562	-0.33262	-0.35087	0.05752	0.22525	0.27510
6	18	2.92266	54.58	3.39615	322.35	3.47430	198.32
		1.69377	2.38182	2.68904	-2.07435	-3.29820	-1.09207
7	8	1.09455	31.89	1.25121	325.18	1.33665	196.78
		0.92932	0.57828	1.02714	-0.71450	-1.27972	-0.38597
7	9	0.23147	32.22	0.23147	272.22	0.23147	152.22
		0.19582	0.12342	0.00897	-0.23130	-0.20480	0.10788
8	19	1.09455	31.89	1.25121	325.18	1.33665	196.78
		0.92932	0.57828	1.02714	-0.71450	-1.27972	-0.38597
9	10	0.39824	75.17	0.39824	315.18	0.39824	195.18
		0.10190	0.38498	0.28245	-0.28074	-0.38436	-0.10424
9	14	0.25174	64.07	0.25174	304.07	0.25174	184.07
		0.11008	0.22640	0.14102	-0.20853	-0.25111	-0.01787
10	11	0.45522	86.09	0.45522	326.09	0.45522	206.09
		0.03103	0.45416	0.37780	-0.25396	-0.40883	-0.20021
13	14	0.30791	273.67	0.30791	153.67	0.30791	33.67
		0.01970	-0.30728	-0.27597	0.13658	0.25626	0.17070

Asymmetric Fault Analysis using X'

LINE CURRENTS

LINE		PHASE A		PHASE B		PHASE C	
		MODULUS	ANGLE	MODULUS	ANGLE	MODULUS	ANGLE
1	5	1.77058	33.01	1.43854	287.89	1.96851	168.15
		1.48471	0.96467	0.44183	-1.36901	-1.92654	0.40434
1	15	4.91744	203.13	8.11069	35.12	9.31530	289.02
		-4.52224	-1.93147	6.63393	4.66630	3.03588	-8.80672
1	20	3.08931	19.78	7.91785	203.70	8.62540	98.46
		2.90704	1.04544	-7.24997	-3.18280	-1.26874	8.53157
2	3	1.48231	24.41	1.35409	315.63	1.59370	190.32
		1.34975	0.61270	0.96799	-0.94687	-1.56790	-0.28561
2	4	1.20945	40.02	1.23660	319.93	1.38084	197.47
		0.92627	0.77768	0.94624	-0.79612	-1.31716	-0.41449
2	5	0.98653	45.07	1.09904	324.18	1.19174	201.54
		0.69669	0.69847	0.89110	-0.64328	-1.10852	-0.43753
2	16	1.54921	348.47	4.81380	343.23	4.19149	252.16
		1.51795	-0.30966	4.60907	-1.38892	-1.28418	-3.98992
2	20	4.97827	201.45	8.27708	152.54	7.36021	43.76
		-4.63351	-1.82037	-7.34476	3.81635	5.31626	5.09018
3	4	0.39484	194.28	0.12982	138.98	0.48073	1.45
		-0.38263	-0.09740	-0.09795	0.08521	0.48058	0.01219
3	17	1.42761	32.97	1.88487	335.60	1.99936	203.36
		1.19769	0.77693	1.71645	-0.77881	-1.83555	-0.79259
4	5	1.00356	205.97	0.63046	104.57	1.07448	350.86
		-0.90220	-0.43952	-0.15864	0.61017	1.06084	-0.17065
4	7	1.03023	49.31	0.94244	318.55	1.38697	186.51
		0.67166	0.78118	0.70637	-0.62388	-1.37802	-0.15730

4	9	0.42499	61.01	0.42499	301.01	0.42499	181.01
		0.20598	0.37174	0.21894	-0.36425	-0.42492	-0.00748
5	6	1.61733	35.89	1.75689	310.95	1.83360	187.60
		1.31030	0.94809	1.15151	-1.32690	-1.81749	-0.24254
6	11	0.36392	271.57	0.36392	151.57	0.36392	31.57
		0.00999	-0.36378	-0.32004	0.17324	0.31004	0.19054
6	13	0.28183	292.49	0.28183	172.49	0.28183	52.49
		0.10781	-0.26040	-0.27941	0.03684	0.17161	0.22356
6	18	2.26068	58.79	2.77074	323.09	2.80708	199.31
		1.17131	1.93357	2.21545	-1.66396	-2.64920	-0.92815
7	8	0.94331	46.73	1.24378	325.78	1.28838	199.13
		0.64656	0.68686	1.02841	-0.69954	-1.21724	-0.42221
8	19	0.94331	46.73	1.24378	325.78	1.28838	199.13
		0.64656	0.68686	1.02841	-0.69954	-1.21724	-0.42221
9	10	0.29172	75.82	0.29172	315.82	0.29172	195.82
		0.07148	0.28283	0.20319	-0.20332	-0.28067	-0.07951
10	11	0.34271	87.63	0.34271	327.63	0.34271	207.63
		0.01416	0.34241	0.28946	-0.18347	-0.30362	-0.15894
13	14	0.23430	275.55	0.23430	155.55	0.23430	35.55
		0.02265	-0.23320	-0.21328	0.09699	0.19063	0.13622

Asymmetric Fault Analysis Using X_s

LINE CURRENTS

LINE		PHASE A		PHASE B		PHASE C	
		MODULUS	ANGLE	MODULUS	ANGLE	MODULUS	ANGLE
1	5	0.72008	49.08	0.72008	289.08	0.72008	169.08
		0.47164	0.54413	0.23541	-0.68052	-0.70705	0.13638
1	15	1.68696	211.57	2.48436	25.98	3.02149	282.27
		-1.43736	-0.88308	2.23331	1.08829	0.64231	-2.95243
1	20	0.97643	29.23	2.61306	190.11	2.89405	87.36
		0.85208	0.47683	-2.57253	-0.45848	0.13330	2.89098
2	3	0.61497	59.31	0.61497	299.31	0.61497	179.31
		0.31385	0.52885	0.30108	-0.53623	-0.61493	0.00737
2	4	0.59805	67.17	0.59805	307.17	0.59805	187.17
		0.23205	0.55120	0.36133	-0.47656	-0.59338	-0.07464
2	5	0.51681	70.30	0.51681	310.30	0.51681	190.30
		0.17422	0.48656	0.33426	-0.39416	-0.50848	-0.09240
2	16	0.52170	312.33	1.75035	337.15	1.62845	248.03
		0.35133	-0.38567	1.61302	-0.67961	-0.60919	-1.51021
2	20	1.69358	201.62	3.50903	146.34	3.21885	37.09
		-1.57443	-0.62401	-2.92065	1.94501	2.56778	1.94099

3	17	0.53846	61.60	0.72317	339.38	0.95828	193.21
		0.25608	0.47367	0.67684	-0.25468	-0.93292	-0.21899
4	5	0.39512	228.72	0.39512	108.72	0.39512	348.72
		-0.26068	-0.29692	-0.12680	0.37422	0.38748	-0.07730
4	7	0.60250	69.42	0.60250	309.42	0.60250	189.42
		0.21183	0.56404	0.38256	-0.46547	-0.59439	-0.09857
4	9	0.21493	64.64	0.21493	304.64	0.21493	184.64
		0.09207	0.19421	0.12215	-0.17684	-0.21422	-0.01737
5	6	0.75756	62.14	0.75756	302.14	0.75756	182.14
		0.35403	0.66975	0.40300	-0.64147	-0.75703	-0.02828
6	18	1.14590	76.07	1.14590	316.08	1.14590	196.07
		0.27578	1.11221	0.82532	-0.79494	-1.10110	-0.31728
7	8	0.70421	77.30	0.70421	317.30	0.70421	197.30
		0.15488	0.68696	0.51749	-0.47761	-0.67237	-0.20935
8	19	0.70421	77.30	0.70421	317.30	0.70421	197.30
		0.15488	0.68696	0.51749	-0.47761	-0.67237	-0.20935

THIS STUDY HAS A MAXIMUM NETWORK TIME OF 3.00 SECONDS

58

```
10     RELAYS
```

5.7.n.1

Relay Data

5.7.n.2

```
 1 IRL A   2 20  1

 1 IRLFB   1

 1 IRLFC   1

 1  CT A   1.0       2.0       1.0       0.167

 1  CTFB   1

 1  CTFC   1

 1 CF1 A  -.44574745    0.74663358     -.42625821     0.1347061

 1 CF2 A  -.24003154E-1 .24017152E-2  -.12566691E-3   .26705756E-5

 1 CF1FB   1

 1 CF1FC   1

 1 CF2FB   1

 1 CF2FC   1
```

```
2 IRL A  1  5  1
2 IRL B  1  5  1
2 IRL C  1  5  1
3 IRL A  5  1  1
3 IRL B  5  1  1
3 IRL C  5  1  1
4 IRL A  5  4  1
4 IRL B  5  4  1
4 IRL C  5  4  1
5 IRL A  4  3  1
5 IRL B  4  3  1
5 IRL C  4  3  1
6 IRL A  3  4  1
6 IRLFB  6
6 IRLFC  6
7 IRL A  4  5  1
7 IRLFB  7
7 IRLFC  7

 8 IRL A  3  2  1
 8 IRLFB  8
 8 IRLFC  8
 9 IRL A  2  3  1
 9 IRLFB  9
 9 IRLFC  9
10 IRL A  1 20  1
10 IRLFB 10
10 IRLFC 10
 2  CT A   1.0      2.0      1.0      0.667
 2  CTFB  2
 2  CTFC  2
 3  CT A   1.0      2.       1.       0.167
 3  CTFB  3
 3  CTFC  3
 4  CT A   1.0      1.0      1.0      0.5
 4  CTFB  4
 4  CTFC  4

 5  CT A   1.0      1.0      1.0      0.333
 5  CTFB  5
 5  CTFC  5
 6  CTFA  4
 6  CTFB  4
 6  CTFC  4
 7  CTFA  5
 7  CTFB  5
 7  CTFC  5
 8  CTFA  3
 8  CTFB  3
 8  CTFC  3
 9  CTFA  2
 9  CTFB  2
 9  CTFC  2
10  CTFA  3
10  CTFB  3
10  CTFC  3
```

2 CF1FA 1
2 CF1FB 1
2 CF1FC 1
3 CF1FC 1
3 CF1FB 1
3 CF1FA 1
4 CF1FA 1
4 CF1FB 1
4 CF1FC 1
5 CF1FC 1
5 CF1FB 1
5 CF1FA 1
6 CF1FA 1
6 CF1FB 1
6 CF1FC 1
7 CF1FC 1
7 CF1FB 1
7 CF1FA 1
8 CF1FA 1
8 CF1FB 1
8 CF1FC 1
9 CF1FC 1
9 CF1FB 1
9 CF1FC 1
10 CF1FA 1
10 CF1FB 1
10 CF1FC 1
2 CF2FA 1
2 CF2FB 1
2 CF2FC 1
3 CF2FA 1
3 CF2FB 1
3 CF2FC 1
4 CF2FA 1
4 CF2FB 1
4 CF2FC 1
5 CF2FA 1
5 CF2FB 1
5 CF2FC 1
6 CF2FA 1
6 CF2FB 1
6 CF2FC 1
7 CF2FA 1
7 CF2FB 1
7 CF2FC 1
8 CF2FA 1
8 CF2FB 1
8 CF2FC 1
9 CF2FA 1
9 CF2FB 1
9 CF2FC 1
10 CF2FA 1
10 CF2FB 1
10 CF2FC 1
//////////
END OF RELAY DATA

Typical Output Available When IPRINT = 1 (See 5.7.a and 6.18)

MOVEMENT	PHASE	MOVEMENT	PHASE	MOVEMENT	PHASE	RELAY NUMBER
0.0000	1	0.0167	2	0.0167	3	1
0.0000	1	0.0026	2	0.0030	3	2
0.0000	1	0.0000	2	0.0000	3	3
0.0000	1	0.0000	2	0.0000	3	4
0.0000	1	0.0000	2	0.0000	3	5
0.0000	1	0.0000	2	0.0000	3	6
0.0000	1	0.0000	2	0.0000	3	7
0.0074	1	0.0000	2	0.0000	3	8
0.0000	1	0.0000	2	0.0000	3	9
0.0167	1	0.0167	2	0.0167	3	10

MOVEMENT	PHASE	MOVEMENT	PHASE	MOVEMENT	PHASE	RELAY NUMBER
0.0000	1	0.0335	2	0.0335	3	1
0.0000	1	0.0051	2	0.0060	3	2
0.0147	1	0.0000	2	0.0000	3	8
0.0335	1	0.0335	2	0.0335	3	10

MOVEMENT	PHASE	MOVEMENT	PHASE	MOVEMENT	PHASE	RELAY NUMBER
0.0000	1	0.9714	2	0.9714	3	1
0.0000	1	0.1047	2	0.1113	3	2
0.9714	1	0.9676	2	0.9713	3	10

MOVEMENT	PHASE	MOVEMENT	PHASE	MOVEMENT	PHASE	RELAY NUMBER
0.0000	1	0.9881	2	0.9881	3	1
0.0000	1	0.1060	2	0.1113	3	2
0.9881	1	0.9836	2	0.9880	3	10

MOVEMENT	PHASE	MOVEMENT	PHASE	MOVEMENT	PHASE	RELAY NUMBER
0.0000	1	1.0049	2	1.0049	3	1
0.0000	1	0.1074	2	0.1113	3	2
1.0019	1	0.9995	2	1.0047	3	10

```
RELAY POSITION   1     PHASE   2     HAS A TRIP IN TIME    0.6100     SECONDS

LINE  2 20     OPENED
LINE  2 20     OPENED
LINE  2 20     OPENED
     0.1005E 01     DISTANCE MOVED BY RELAY   1     PHASE   2

RELAY POSITION   1     PHASE   3     HAS A TRIP IN TIME    0.6100     SECONDS


RELAY POSITION  10     PHASE   1     HAS A TRIP IN TIME    0.6100     SECONDS

LINE  1 20     OPENED
LINE  1 20     OPENED
LINE  1 20     OPENED
     0.1005E 01     DISTANCE MOVED BY RELAY  10     PHASE   1

RELAY POSITION  10     PHASE   3     HAS A TRIP IN TIME    0.6100     SECONDS

                                   THE TIME VALUES ARE  0.610000  1.805000  3.000000
20     3 FBUS    FAULT

FAULT ISOLATED.  RUN STOPPED
```

CHAPTER 3

RELAY CHARACTERISTICS AND CURVE FITTING

3.1 General

This work is based on an approach which uses the system conditions, as computed at three significant values of time - initially the times will be related to T" and T' of one of the machines within the system, to determine the behaviour of the system and the performance of the protection applied to that system. In order that this objective may be attained, it is necessary to know the behaviour of the system at times other than those specified above. This is achieved by generating a series of approximating functions which will allow currents, voltages and machine impedances to be evaluated at any required time.

The approximating functions for the currents and voltages are used with a further set of functions which represent the relay characteristics to reproduce the protection system performance.

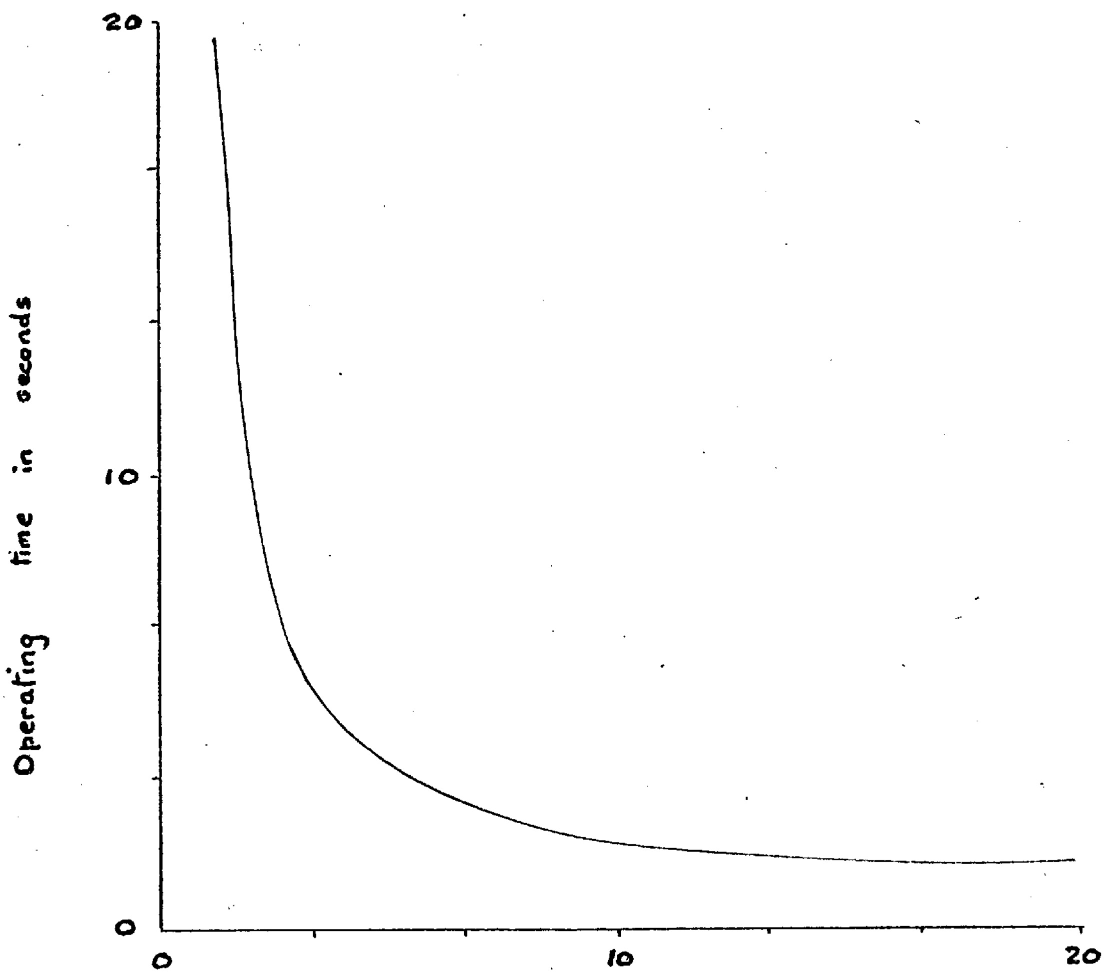
3.2 Relay Characteristics

Nearly all overcurrent relays in use at the present time are induction disc relays, and although transistorised relays are increasing in number, they will only form a very small percentage of the total for the next decade or so. For this reason, the characteristics considered here are those of induction disc relays.

3.2.1 Overcurrent Relay Characteristics (15,16)

All induction disc relays have the same form of characteristic, typically as shown by Fig. 3.1.

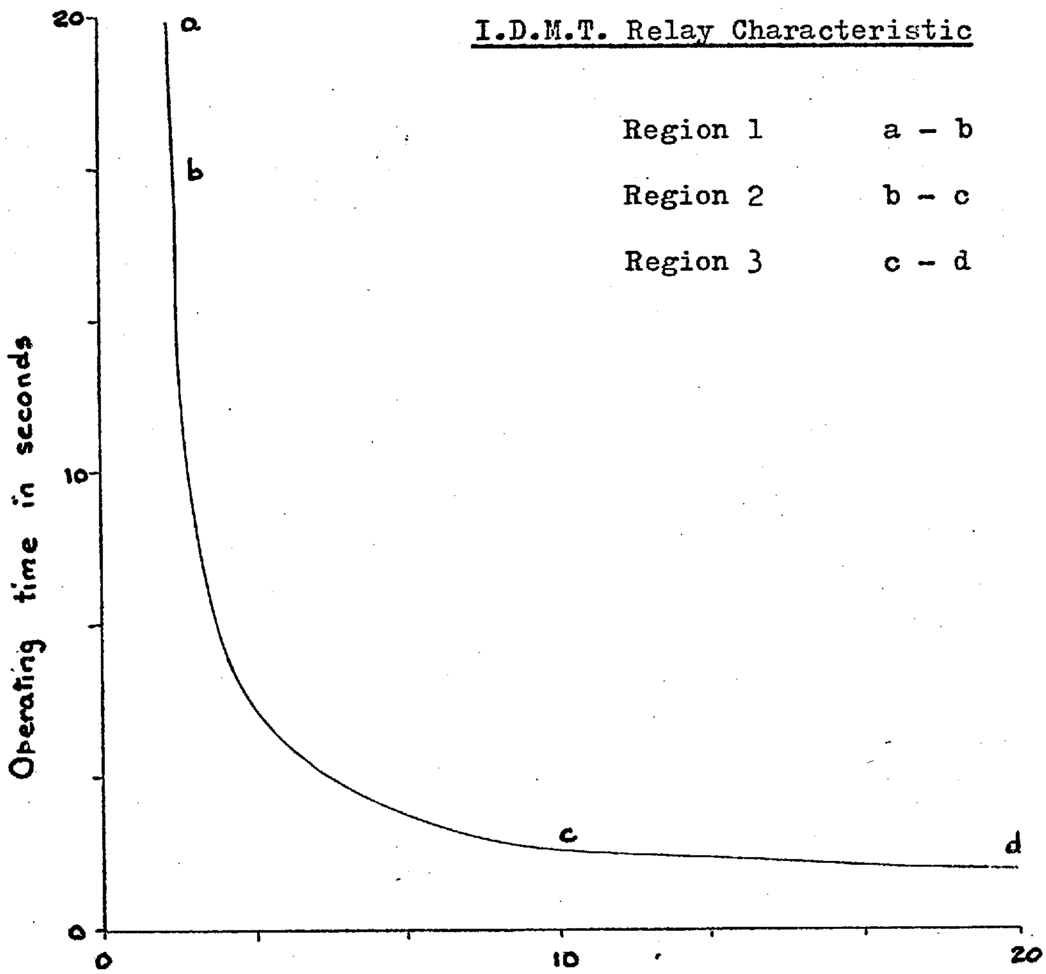
I.D.M.T. Relay Characteristic



Multiples of plug setting

Fig. 3.1

This type of characteristic is known as an Inverse Definite Minimum Time (I.D.M.T.) characteristic. The characteristic can be divided into three distinct regions, as shown by Fig. 3.2.



Multiples of plug setting

Fig. 3.2

Region 1. At low values of current the characteristic is determined by the effect of the restraining forces of the control spring.

Region 2. This is the inverse section of the characteristic, and the relay is designed so that $t = k/I$ in this region.

Region 3. The region of definite minimum time. The relay performance on this section of its characteristic is determined by the magnetic saturation effects caused by the high values of current.

The time ordinate should, theoretically, be proportional to the time multiplier setting, but this is not possible at low current values because ofthe inertial properties of the disc. The error caused by this effect is known as 'pick up' error. When the disc is moving it will continue to move, owing to its momentum, after the current is switched off. It is, therefore, possible for a relay to operate after a fault has been cleared. This type of error is known as 'overshoot'.

The present U.K. practice is that these errors should be within the limits specified by B.S. 142. This means that the same characteristic curve may be used for any time multiplier setting and the relay performance will be within the B.S. 142 specification.

There are two major constraints on the functions which are used to generate these relay characteristics.

1. They must have sufficient accuracy to comply with the B.S. 142 specification

and 2. they should have a minimum of coefficients, consistent with (1) above; this reduces computer storage requirements and, subsequently, computing time.

3.2.2 Previous Work

Previous workers (17,18,19,21,23) have used three approaches to this problem.

1. A table look-up approach used by Graham and Watson (17), where the device characteristics are stored in table form, and some form of interpolation is used for intermediate values. This satisfies (1) above but not (2).

2. Special equations developed by Heiber (18), satisfy the accuracy constraint, but do not minimise the computation time, and are also difficult to produce.

3. A special type of polynomial fit, used by Albrect (19), requiring 20 constants for each relay curve.

3.2.3 A New Approach to Relay Characteristic Approximation

The shape of the I.D.M.T. relay characteristic is of the same form as a curve which is generated by a simple rational function $R(x)$, where $R(x) = 1.0/(b_1+b_2.x)$. The method proposed, uses a rational function to generate the relay characteristics. A method of determining the coefficients of an adequate function of this form is described in the following paragraphs.

3.3 Rational Functions (25)

A rational function is a function which has the form

$$R(x) = \frac{a_1 + a_2x + a_3x \ \ldots}{b_1 + b_2x + b_3x \ \ldots} \qquad (3.1)$$

For reasons of simplicity an initial approximation is derived, in which $a_1 = 1$ and $a_2 = a_3 = a_n = 0$

Thus

$$R(x) = \frac{1}{b_1 + b_2x + b_3x \ \ldots} \qquad (3.2)$$

3.3.1 The Standard Method (25)

The standard method of obtaining a rational function approximation to a set of given data is as follows: for example, given the following data

$x_1 = 1.0$	$x_2 = 2.0$	$x_3 = 3.0$
$y_1 = 10.0$	$y_2 = 5.5$	$y_3 = 4.667$

(from the function $y = \frac{6}{x^2} + 4$)

an equation is found, such that it generates a curve which misses the specified data points by $\pm$ h. If h has the minimum possible value, the approximating function is known as a mini-max rational function approximation, or as a Chebyshev approximation.

We have, therefore,

$$y_k = \frac{1}{b_1 + b_2 x_k} + (-1)^{k+1}.h \qquad (3.3)$$

for a first order rational function approximation. Thus,

$$y_1 = 10.0 = \frac{1}{b_1 + b_2 x_1} + h \qquad (3.4.1)$$

$$y_2 = 5.5 = \frac{1}{b_1 + b_2 x_2} - h \qquad (3.4.2)$$

$$y_3 = 4.667 = \frac{1}{b_1 + b_2 x_3} + h \qquad (3.4.3)$$

rearranging, we have

$$b_1 (y_1 - h) + b_2(y_1 - h)x_1 = 1 \qquad (3.5.1)$$

$$b_1 (y_2 + h) + b_2(y_2 + h)x_2 = 1 \qquad (3.5.2)$$

$$b_1 (y_3 - h) + b_2(y_3 - h)x_3 = 1 \qquad (3.5.3)$$

These equations form a non-linear set in b_1 , b_2 and h.

Using Cramer's Rule

$$\begin{vmatrix} (y_1 - h) & (y_1 - h)x_1 & -1 \\ (y_2 + h) & (y_2 + h)x_2 & -1 \\ (y_3 - h) & (y_3 - h)x_3 & -1 \end{vmatrix} = 0 \qquad (3.6)$$

Substituting the values for x_k and y_k equation (3.6) becomes

$$4h^2 - 33.1h + 12.66 = 0 \qquad (3.7)$$

from which $h = 4.15 \pm 3.74$

and taking the root with the smallest modulus gives

$$h = 0.41$$

This value of h is then substituted into any two of equations (3.4), say (3.4.1) and (3.4.2) giving

$$b_1(9.6) + b_2(9.6) = 1$$

$$b_1(5.9) + b_2(5.9) = 1$$

hence $b_1 = 0.039$ and $b_2 = 0.065$

and the approximating function is

$$y = \frac{1}{0.039 + 0.065x} \qquad (3.8)$$

Equation (3.8) is plotted with the original data function in Fig. 3.4.

When more than three data points are given the 'exchange' procedure is adopted, that is, the values of b_1, b_2 and h are found using the first three data points. The differences between the given values of y and those generated by the prediction equation are then computed. The data pair giving the greatest difference (or error) is then used to replace one of the original data pairs, such that the correct sign of h is maintained. New values of b_1, b_2 and h are then computed.

Convergence is rapid, and an example with 30 data pairs required only three exchanges to determine the Chebyshev fit. (A Chebyshev fit is obtained when the error at each data point is $\pm h$).

The great disadvantage of this method is that a determinental equation has to be solved in order to obtain a value for h. When the order of the determinent is greater than 4 this is very difficult. However, it is possible to evaluate the determinent using various values of h, until a value is found that gives $\Delta = 0$. I have not investigated this approach because it is more complex than my proposed method, and the advantages it offers are very small.

3.3.2 An Alternative Approach

The method of solution which is proposed by the author is as follows:

1) A value for h is specified in equation (3.5)

2) The over-determined set of equations in b_1 and b_2 are solved using the method of least squares.

3) The values of b_1 and b_2 obtained from (2) above are substituted in equations (3.5) which are solved by the method of least squares for an improved value of h.

4) The procedure is repeated until satisfactory convergence is obtained.

For example, using the data of the previous example and setting the initial value of h to zero, equations (3.5) become:

$$10b_1 + 10b_2 - 1 = 0$$
$$5.5b_1 + 11b_2 - 1 = 0$$
$$4.667b_1 + 14b_2 - 1 = 0$$

Solving by the method of least squares

$$b_1 = 0.0407, \qquad b_2 = 0.06185$$

substituting these values into (3.5), the method of least squares yields a new value for h, $h_{new} = 0.025$.
Repeating the process

$$b_1 = 0.0396, \qquad b_2 = 0.0640$$

correct to 4 decimal places. Subsequent iterations make changes in the 6th decimal place.

If further terms are added to the rational function,

$$R(x) = \frac{1}{b_1 + b_2x + b_3x^2 \;\ldots}$$

a better approximation is obtained. Experience has shown that functions having 8 or 9 terms usually provide minimum error.

3.4 A Program for obtaining a Least Squares Approximation to a Chebyshev Rational Function Approximation to a set of given data points.

3.4.1 Summary

This program takes a set of ordered data and computes the coefficients of a first order rational function approximation. A second order function is then found, if the mean square error is reduced by this second function, a third order function is determined. This procedure is repeated until an increase in the order of rational function does not reduce the mean square error. The coefficients of the rational function giving a minimum value for the mean square error are printed as results, see 3.13.

3.4.2 To obtain an mth Order RF Approximation

For an mth order RF we have, with $a_1 = 1$, $a_2 = a_3 = 0$ etc.

$$y_k = \frac{1}{b_1 + b_2x_k + b_3x_k + \ldots b_jx_k} + (-1)^{k+1}.h \qquad (3.9)$$

where $j = (m + 1)$

Rearranging, and letting

$c_1 = (y_1 - h)$, $c_2 = (y_2 + h)$, $c_3 = (y_3 - h)$ etc

then,

$$\begin{bmatrix} c_1 & c_1x_1 & c_1x_1 & \cdots & c_1x_1 \\ c_2 & & & & \\ \cdot & & & & \\ \cdot & & & & \\ c_n & c_nx_n & \cdots & & c_nx_n \end{bmatrix} \begin{bmatrix} b_1 \\ \cdot \\ \cdot \\ \cdot \\ b_n \end{bmatrix} = \begin{bmatrix} 1 \\ \cdot \\ \cdot \\ \cdot \\ 1 \end{bmatrix} \qquad (3.10)$$

That is $\underline{C}.\underline{b} = \underline{B}$

A least squares solution to equations (3.10) is obtained as follows:

1. Enlarge $\underline{C}$ by adding $\underline{B}$ as the (m+1)th column, to form a new matrix $\underline{A}$.

2. Compute $\underline{R} = \underline{A}^T.\underline{A}$. $\underline{R}$ is $(m+1)x(m+1)$

3. Remove the last row and column of $\underline{R}$ to form $\underline{S}$. $\underline{S}$ is $(m)x(m)$.

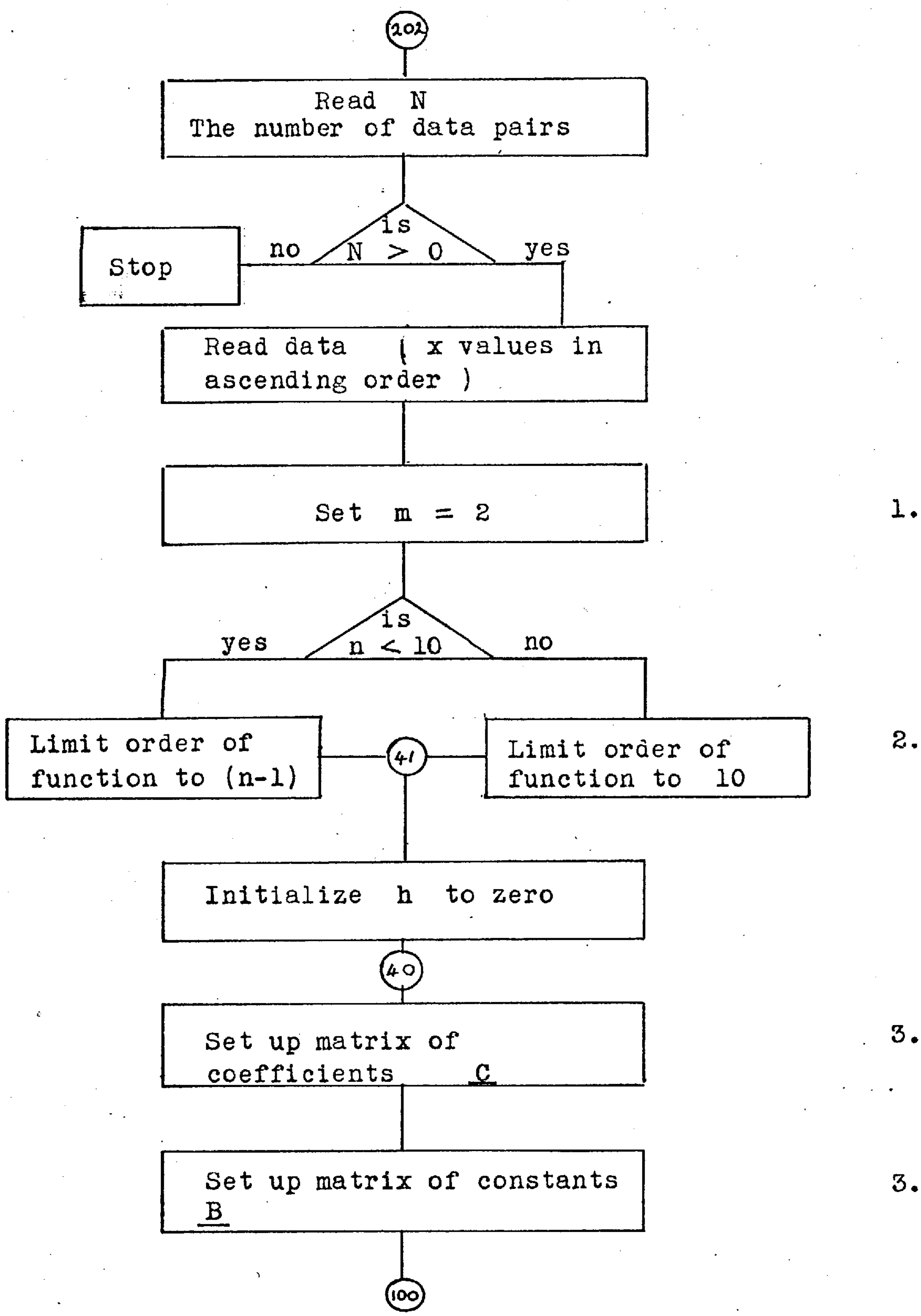
4. The first m elements of the last column of $\underline{R}$ form a vector $\underline{D}$.

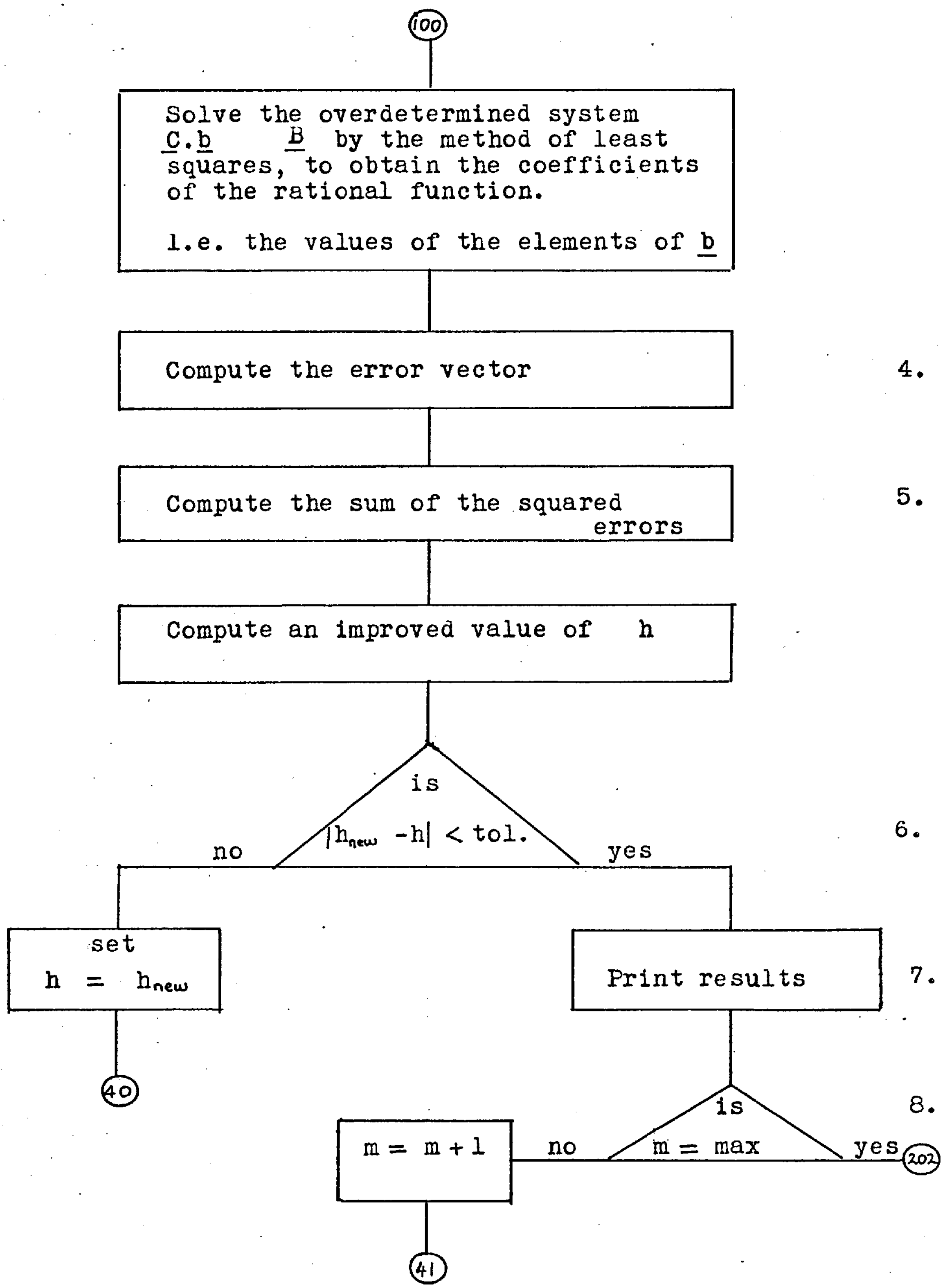
5. Solve the m simultaneous equations

$$\underline{S} \,.\underline{b} = \underline{D} \qquad (3.11)$$

This completes the solution by the method of least squares, and the vector $\underline{b}$ contains the required coefficients of the rational function.

3.4.3 Program Flow Chart




100
Solve the overdetermined system C.b B by the method of least squares, to obtain the coefficients of the rational function.
i.e. the values of the elements of b
Compute the error vector
4.
Compute the sum of the squared errors
5.
Compute an improved value of h
is |h_new - h| < tol.
no
yes
6.
set h = h_new
40
Print results
7.
is m = max
8.
no
yes
202
m = m + 1
41

1. m is the number of coefficients in the rational function. m starts with a value of 2 i.e. the first function found is

$$y = \frac{1}{b_1 + b_2 x}$$

2. If there are less than 10 data points, the maximum value of m is limited to (n - 1), otherwise m is limited to 10. That is, the maximum power of x in the rational function is 9.

3. Equations (3.10)

4. The error vector E is computed from

$$E_k = y_k - 1.0/(b_1 + b_2 x_k + b_3 x^2_k \quad \ldots \quad)$$

5. The sum of the squared errors $HSQ = \sum_{k=1}^{N} E^2_k$
This is computed for each rational function, and if a function of order (j+1) has a greater HSQ than the function of order j, computation stops and the next set of data is read in.

6. The iteration procedure stops when two consecutive values of h are within the tolerance, (0.0001).

7. See typical output, sections 3.5.1 and 3.13.

8. At this point comparison of the sum of the squared errors is made. If the new value of HSQ is less than the previously computed value, m is increased by one, otherwise the program returns to program statement number 202.

3.5 Results

Fig. 3.3 shows a very inverse type of relay characteristic. On the scale of this diagram the 4th order rational function approximation cannot be distinguished from the original data curve. For comparison, the optimum (5th order) polynomial approximation is also shown. This type of polynomial approximation is completely inadequate for relay characteristic approximation since the errors are relatively large - 100%, and also of alternating in sign.

Comparison of Relay Characteristic and Approximations

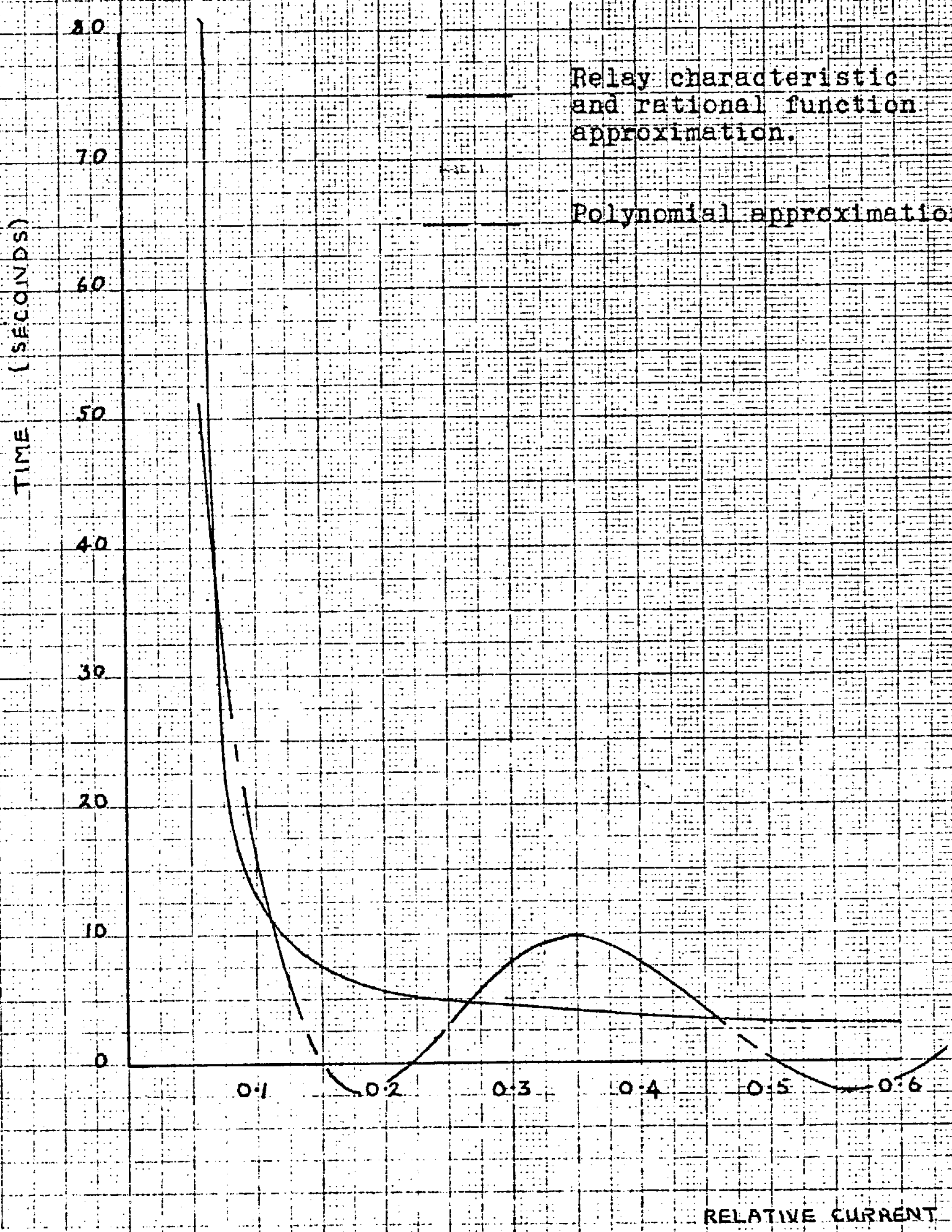


Fig. 3·3

3.5.1 Computer output for very inverse (Type TJX) Relay Characteristic Approximation

The most accurate approximation for this relay characteristic was obtained from a function having six coefficients, as follows:

b_1 =	− 0.164270	b_2 =	3.938397
b_3 =	− 18.768524	b_4 =	48.478347
b_5 =	− 58.841201	b_6 =	26.629024

Characteristic values correct to two decimal places are:

Actual Current (Amperes)	Time (Seconds)	Estimated Time (Seconds)	Error (p.u.)
3.0	76.5	70.44	-0.08
3.5	29.54	28.81	-0.25
4.0	18.02	18.80	0.04
4.5	14.60	14.33	-0.02
5.0	12.50	11.80	-0.06
6.0	9.25	9.06	-0.02
7.0	7.75	7.62	-0.02
8.0	6.85	6.74	-0.02
9.0	6.20	6.15	-0.01
10.0	5.72	5.71	-0.00

3.6 Exponential Approximations

3.6.1 Preamble

Currents and voltages of electric power systems tend to decay in an exponential manner, also, the shape of the relay characteristics previously discussed are not dissimilar to exponential curves. For these reasons exponential approximations were investigated.

3.6.2 Prony's Method (26)

Let $f(x)$ be a function which is tabulated at equal intervals $f(0)$, $f(1)$ ... $f(2n-1)$.

Then, in general, for an exponential approximation

$$\begin{aligned} f(x) &= \sum_{k=0}^{n} C_k \exp(\lambda_k x) \\ &= \sum_{k=0}^{n} C_k \mu_k^x \end{aligned} \tag{3.12}$$

where C_k, λ_k and hence μ_k are unknown real parameters which have to be determined. The problem is non-linear, but a solution may be obtained by the use of Prony's method, as follows:

Let μ_1, μ_2, .. μ_n be the roots of the algebraic equation

$$\mu^n + \alpha_1 \mu^{n-1} + \quad \dots \quad \alpha_n = 0 \tag{3.13}$$

Then

$$f(n) + \alpha_1 f(n-1) \quad \ldots \quad \alpha_n f(0) = 0$$

$$f(n+1) + \alpha_1 f(n) \quad \ldots \quad \alpha_n f(1) = 0$$

$$\cdot$$

$$\cdot \qquad (3.14)$$

$$\cdot$$

$$f(2n-1) + \alpha_1 f(2n-2) \quad \ldots \quad \alpha_n f(n-1) = 0$$

The roots $\mu_1, \mu_2 \cdots \mu_n$ of equation (3.13) are computed and then any n of the linear equations (3.12) are solved for the coefficients C_k.

For example, using the data of the previous illustrative example on page

$x_1 = 1 \qquad x_2 = 2 \qquad x_3 = 3$

$y_1 = 10 \qquad y_2 = 5.5 \qquad y_3 = 4.67$

From (3.14)

$$f(1) + \alpha_1 f(0) = 0$$

$$f(2) + \alpha_1 f(1) = 0$$

(Note the x values have been transformed to (x-1))

Therefore,

$$5.5 + \alpha_1 10 = 0$$

$$4.67 + \alpha_1 5.5 = 0$$

solving by least squares $\alpha_1 = -0.62$

and hence $\mu - 0.62 = 0$, therefore, $\mu = 0.62$

From (3.12) $\mu = \exp(\lambda)$, thus

$$10 = 0.62^0 .C_1$$

$$5.5 = 0.62^1 .C_1$$

$$4.67 = 0.62^2 .C_1$$

and again by least squares, $C_1 = 9.92$. Thus a first approximation is $y = 9.92 \mu^x$ or $y = 9.92.\exp(-0.475x)$. This function is plotted on Fig.3.4 with the original data function and the rational function approximation, so that a direct comparison may be made. Typical results, obtained by the use of this method are shown in section 3.13.

Comparison of Rational Function and Exponential Approximations

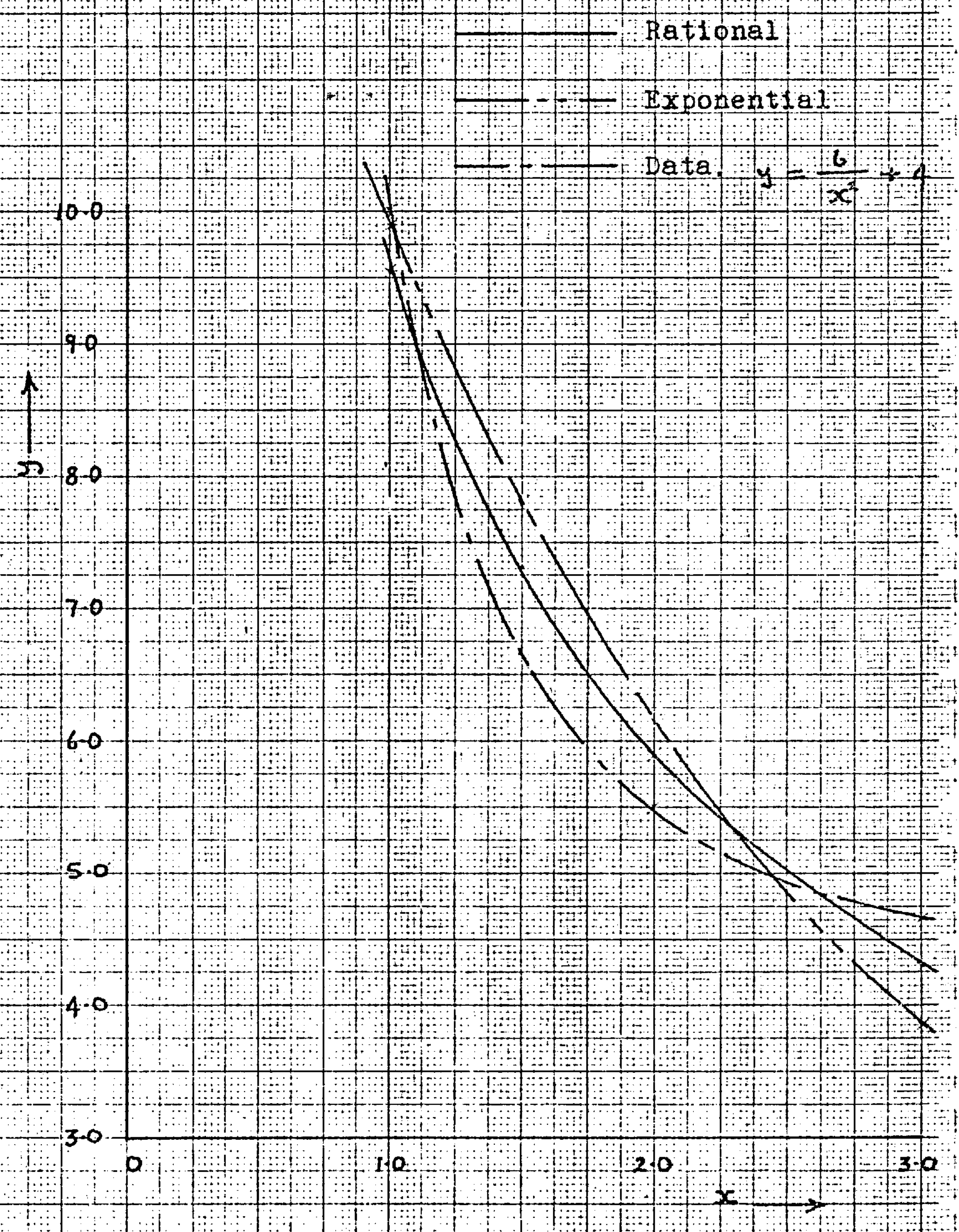


Fig. 3.4

3.7 Exponential Approximation 2

The disadvantages of the exponential approximations produced from Prony's method made it necessary to investigate other methods of producing, if possible, more general exponential approximations. It is reasonable to assume that the currents and voltages of an electrical power system will decay in terms of a function having two coefficients, i.e.

$$I(t) = A.\exp(\alpha_1 t) + B.\exp(\alpha_2 t) \qquad (3.15)$$

Then

$$I_1 = I(0) = A + B \qquad (3.15.1)$$

$$I_2 = I(t_1) = A.\exp(\alpha_1 . t_1) + B.\exp(\alpha_2 . t_1) \qquad (3.15.2)$$

$$I_3 = I(t_2) = A.\exp(\alpha_1 . t_2) + B.\exp(\alpha_2 . t_2) \qquad (3.15.3)$$

Let $\exp(\alpha_1) = Z_1$ and $\exp(\alpha_2) = Z_2$ (3.16)

hence

$$I_1 = A + B \qquad (3.16.1)$$

$$I_2 = A.Z_1^{t_1} + B.Z_2^{t_1} \qquad (3.16.2)$$

$$I_3 = A.Z_1^{t_2} + B.Z_2^{t_2} \qquad (3.16.3)$$

rearranging equations (3.16.1) and (3.16.2) and substituting into (3.16.3) gives:

$$I_3 = A.Z_1^{t_2} + (I_1 - A).\left[\frac{I_2 - A.Z_1^{t_1}}{I_1 - A}\right]^{(t_2/t_1)} \qquad (3.17)$$

Note: If a 'real' solution is to be found $Z_2 = \exp(\alpha_2)$ must be positive, and if $A > I_1$ then $A.Z_1 > I_2$, or if $A < I_1$ then $A.Z_1 < I_2$.

Equation (3.17) can be solved for a given value of A, to determine Z_1, the values of B and Z_2 are then found from (3.16.2) and (3.16.1.)

For example if

$$I(0) = 10$$
$$I(0.9) = 5$$
$$I(1.9) = 2$$

Choose any reasonable value for A, say 10.5, then from equation (3.17), $Z_1 = 0.50$ (approximately). Any method which is suitable for non-linear equations may be used to obtain the value of Z_1. Hence $Z_2 = 1.27$ and $B = -0.5$. The equation thus obtained is:

$$I_k = 10.5\exp(-0.693t_k) - 0.5\exp(-0.239t_k)$$

This method has the advantage that the values of the independent variable do not have to be equally spaced, as for Prony's method.

3.8 Exponential Approximation 3

It is possible to use the above method to derive a more simple exponential approximation

$$I(t) = A + B.\exp(\alpha.t) \tag{3.18}$$

However, experience has shown that equations (3.15) and (3.17) have a serious disadvantage in that the computation of the coefficients has to be made to an accuracy of at least 30 decimal places. It was therefore decided that a more stable function would be used to approximate the current and voltage decay.

3.9 An Alternative Approach

The exponential approximating functions have proven to be too unstable for practical usage. The coefficients A and B take values between $\pm 10^{20}$ for typical data values, and it is not possible to obtain sufficient accuracy when differences are required, i.e. $A - B.e^{\alpha t}$

Intuitively, the following function was chosen

$$I(t) = A + B.t + C.t^{\frac{1}{2}} \tag{3.19}$$

The coefficients of(3.18) are easily determined, since the values of the currents and voltages are always available at three distinct values of time.

Equation (3.18) is, therefore, used to form three linear simultaneous equations from which the values of A, B and C are determined.

For comparison, the following table lists the values of two exponential functions and the corresponding results obtained from the approximating functions.

Function $f_1 = 10.\exp(-0.3t) + 2.\exp(0.2t)$

Function $f_2 = 2.\exp(-t) + 1.\exp(-2t)$

Time	f_1	(3.18)	f_2	(3.18)
0.0	12.000	12.000	2.000	2.000
0.18	11.404	11.224	1.532	1.288
0.36	10.837	10.704	1.184	1.032
0.54	10.299	10.226	0.922	0.850
0.72	9.789	9.769	0.723	0.706
0.90	9.304	9.326	0.572	0.588
1.08	8.844	8.893	0.455	0.487
.				
.				
1.71	7.407	7.427	0.214	0.222

With this function there are no stability problems and an exact fit is obtained at the three data points.

3.10 Impedance Relays

The impedance relay model used in this work has a typical three zone characteristic, with directional facility.

For example

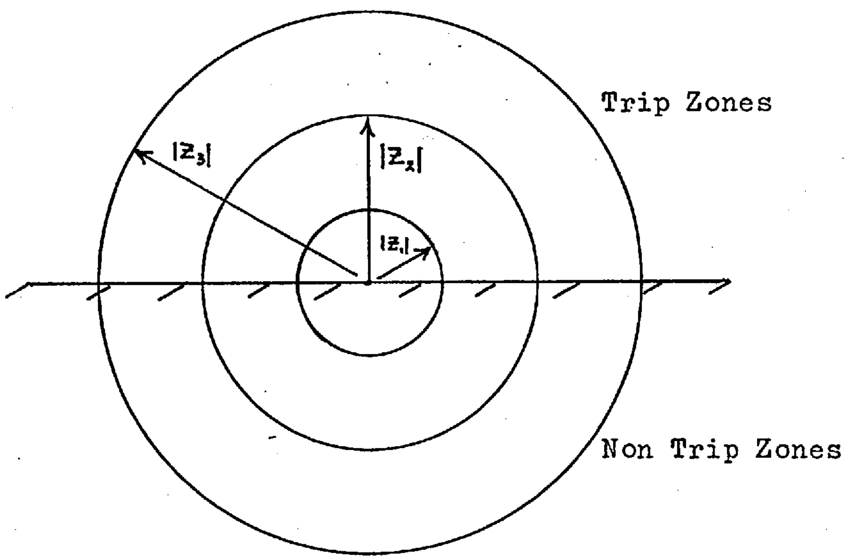


Fig. 3.5

The following data is therefore required for each relay:

1. $|Z|$ for each zone
2. Trip time for each zone
3. Directional indication, if any.

3.10.1 Earth Fault Detection

The method adopted for earth fault detection is known as Residual Current Compensation. The impedance equation[30] is, therefore,

$$Z_1 = \frac{V_{AE}}{I_A + I_O(\frac{Z_o}{Z_1} - 1)}$$

$$= \frac{V_{AE}}{I_A + (I_A+I_B+I_C)(\frac{k-1}{3})} \qquad (3.20)$$

where Z_1 is the impedance seen by the relay and

$$k = Z_o/Z_1$$

For reasons of simplicity the present program uses a constant value of 2.0 for k.

2.10.2 Phase to Phase Faults

For phase to phase faults the voltage across the two faulty phases and the phasor difference of the currents in the faulty phases are applied to the relay. For a fault on phases B and C, the relay will see

$$Z = \frac{V_{BC}}{I_B - I_C} \qquad (3.21)$$

3.11 Determination of I.D.M.T. Relay Movements

As has been previously shown in this chapter it is possible to determine equations for the system currents, and for the relay characteristics. The following procedure uses these equations to calculate the movement of the relay contacts during any given time interval. The relay characteristic is a function of current,

$$T = f_1(I)$$

where T is the time taken for the relay contacts to close. The relay current is also a function time,

$$I = f_2(t)$$

hence,

$$T = f_1(f_2(t)) = f_3(t)$$

The distance moved, DM, by the contacts during a small interval dt is given by,

$$DM = \frac{dt}{f_3(t)} \quad \text{per unit.}$$

Therefore, the total distance moved by the relay contacts during an interval τ is

$$DM = \int_0^{\tau} \frac{dt}{f_3(t)} \qquad (3.22)$$

For example, if $f_1(I) = \frac{1}{a + bI}$

and $f_2(t) = A\exp(-\alpha t)$, the distance moved during an interval t_1 to t_2 is given by

$$DM = \int_{t_1}^{t_2} (a + c.\exp(-\alpha t)dt$$

This program uses Simpson's Rule to evaluate this integral, and the functions F1, F and F2 to evaluate $f_3(t)$ for the single line to earth, line to line and double line to earth faults respectively, see 6.43.

3.12 Determination of Impedance Relay Performance

Equations (3.20) and(3.21) are used, depending on the type of fault, to evaluate the impedance seen by the relay. This value of impedance is compared with the relay characteristic, impedance and time values, and the appropriate action is taken, see 6.19.

3.13 Listed Relay Characteristics - Actual and Approximations

The following lists included the coefficients of the approximating functions, the specified function values and the estimated values.

3.13.1

DATA FIT BY RATIONAL FUNCTION OF THE FORM 1/(A + BX + CX**2 + DX**3 ECT ...)

'TJX' RELAY WITH 3 SECOND DEFINITE MINIMUM TIME.

COEFFICIENT	POWER OF X
-0.55632614E 00	0
0.10234249E 01	1
-0.68616766E 00	2
0.24943473E 00	3
-0.47790230E-01	4
0.37373325E-02	5
0.21177355E-03	6
-0.65744463E-04	7
0.48321210E-05	8
-0.12358829E-06	9

THE SUM OF THE SQUARED ERRORS IS 0.367500E 02

THE RMS ERROR IS 0.112572E 01

X VALUE	Y VALUE	Y ESTIMATE	ERROR	P.U. ERROR
1.100000	138.000000	133.646390	0.435361E 01	0.315479E-01
1.120000	85.000000	88.568228	-0.356823E 01	-0.419791E-01
1.160000	54.000000	53.921400	0.785997E-01	0.145555E-02
1.180000	44.000000	45.454626	-0.145463E 01	-0.330597E-01
1.200000	38.000000	39.456657	-0.145666E 01	-0.383331E-01
1.260000	29.000000	28.774245	0.225755E 00	0.778467E-02
1.320000	23.500000	23.094485	0.405515E 00	0.172559E-01
1.400000	19.000000	18.688596	0.311404E 00	0.163897E-01
1.500000	16.000000	15.473531	0.526469E 00	0.329043E-01
1.750000	12.200000	12.107007	0.929925E-01	0.762234E-02
2.000000	9.800000	9.747186	0.528139E-01	0.538917E-02
2.500000	7.600000	7.827735	-0.227735E 00	-0.299651E-01
3.000000	6.500000	6.657298	-0.157298E 00	-0.241997E-01
3.500000	5.800000	5.834640	-0.346397E-01	-0.597236E-02
4.000000	5.400000	5.268978	0.131022E 00	0.242633E-01
4.500000	5.000000	4.896268	0.103732E 00	0.207464E-01
5.000000	4.600000	4.649919	-0.499186E-01	-0.108519E-01
5.500000	4.400000	4.467880	-0.678805E-01	-0.154274E-01
6.000000	4.200000	4.304120	-0.104120E 00	-0.247904E-01
6.500000	4.100000	4.139400	-0.394003E-01	-0.960984E-02
7.000000	4.000000	3.982685	0.173148E-01	0.432870E-02
7.500000	3.900000	3.858837	0.411629E-01	0.105546E-01
8.000000	3.800000	3.789517	0.104832E-01	0.275872E-02
8.500000	3.750000	3.775000	-0.250004E-01	-0.666678E-02
9.000000	3.700000	3.778823	-0.788233E-01	-0.213036E-01
9.500000	3.650000	3.725770	-0.787702E-01	-0.215809E-01
10.000000	3.600000	3.576903	0.230975E-01	0.641596E-02
10.500000	3.550000	3.428161	0.121839E 00	0.343207E-01
11.000000	3.500000	3.751323	-0.251323E 00	-0.718065E-01

DATA FIT BY RATIONAL FUNCTION OF THE FORM 1/(A + BX + CX**2 ETC)

3.13.2

RATIONAL FUNCTION APPROXIMATION, TYPE 'CDG' RELAY

COEFFICIENT	POWER OF X
0.021962	0
-0.052511	1
0.039642	2
-0.001767	3
-0.000166	4
0.000018	5
-0.000000	6

THE SUM OF THE SQUARED ERRORS IS 0.864394E 00 H IS 0.200000

X VALUE	Y VALUE	Y ESTIMATE	ERROR	P.U. ERROR
1.500000	38.750007	38.860099	-0.110092E 00	-0.284110E-02
2.000000	17.500003	16.871154	0.628847E 00	0.359341E-01
2.500000	9.250001	9.433944	-0.183943E 00	-0.198857E-01
3.000000	5.650000	6.094563	-0.444563E 00	-0.786837E-01
3.500000	4.200000	4.315878	-0.115878E 00	-0.275900E-01
4.000000	3.250000	3.256104	-0.610466E-02	-0.187835E-02
4.500000	2.650000	2.572919	0.770804E-01	0.290869E-01
5.000000	2.200000	2.106093	0.939064E-01	0.426847E-01
5.500000	1.800000	1.772511	0.274883E-01	0.152712E-01
6.000000	1.550000	1.525535	0.244642E-01	0.157833E-01
6.500000	1.330000	1.337328	-0.732852E-02	-0.551017E-02
7.000000	1.150000	1.190419	-0.404193E-01	-0.351472E-01
7.500000	1.000000	1.073380	-0.733806E-01	-0.733806E-01
8.000001	0.900000	0.978470	-0.784703E-01	-0.871892E-01
8.500001	0.800000	0.900288	-0.100288E 00	-0.125360E 00
9.000001	0.740000	0.834973	-0.949737E-01	-0.128342E 00
9.500001	0.700000	0.779705	-0.797054E-01	-0.113864E 00
10.000001	0.680000	0.732390	-0.523903E-01	-0.770446E-01
10.500001	0.640000	0.691455	-0.514551E-01	-0.803986E-01
11.000001	0.600000	0.655710	-0.557101E-01	-0.928503E-01
11.500001	0.570000	0.624255	-0.542555E-01	-0.951852E-01

DATA FIT BY RATIONAL FUNCTION OF THE FORM 1/(A + BX + CX**2 ETC)

3.13.3

COEFFICIENT	POWER OF X
0.046784	0
-0.131860	1
0.137057	2
-0.059438	3
0.017472	4
-0.003986	5
0.000595	6
-0.000038	7

RATIONAL FUNCTION APPROXIMATION.

0·5 Ampere THERMAL RELAY.

THE SUM OF THE SQUARED ERRORS IS 0.289637E 01 H IS 0.200000

X VALUE	Y VALUE	Y ESTIMATE	ERROR	P.U. ERROR
1.000000	152.000030	151.835541	0.164473E 00	0.108205E-02
1.250000	79.200012	79.364120	-0.164114E 00	-0.207215E-02
1.500000	48.500007	47.447563	0.105244E 01	0.216998E-01
1.750000	31.500003	32.438751	-0.938750E 00	-0.298015E-01
2.000000	24.000003	24.409385	-0.409383E 00	-0.170576E-01
2.250000	19.400001	19.657280	-0.257279E 00	-0.132618E-01
2.500000	16.800003	16.623001	0.176999E 00	0.105356E-01
2.750000	15.000001	14.559970	0.440030E 00	0.293353E-01
3.000000	13.400001	13.066829	0.333171E 00	0.248635E-01
3.250000	12.000001	11.905107	0.948926E-01	0.790772E-02
3.500000	10.800001	10.923978	-0.123977E 00	-0.114794E-01
3.750000	9.600000	10.030151	-0.430152E 00	-0.448075E-01
4.000000	9.000001	9.178117	-0.178117E 00	-0.197908E-01
4.250000	8.400001	8.364963	0.350362E-01	0.417098E-02
4.500000	7.800000	7.621826	0.178173E 00	0.228427E-01
4.750000	7.100000	7.003091	0.969076E-01	0.136489E-01
5.000000	6.600000	6.583710	0.162892E-01	0.246806E-02

3.13.4 Exponential Approximation (Prony's Method) for 0.5 A Thermal Relay

Approximating function

$$t = -0.219086 \ \text{Exp}\ (-1.62335\ I)$$
$$-1.885370 \ \text{Exp}\ (-0.898015\ I)$$
$$56.65500 \ \text{Exp}\ (-0.807749\ I)$$
$$24.646800 \ \text{Exp}\ (-0.090041\ I)$$

I	t	t Estimate	P.U. Error
0.0000	79.2	79.1974	0.00003
1.0	48.5	46.9735	0.0315
2.0	31.5	31.5262	-0.0008
3.0	24.0	23.7049	0.0123
4.0	19.4	19.3793	0.0010
5.0	16.8	16.6893	0.0066
6.0	15.0	14.7958	0.0136
7.0	13.4	13.3178	0.0061
8.0	12.0	12.0800	-0.0067
9.0	10.8	10.9999	-0.0184
10.0	9.6	10.0339	-0.0452
11.0	9.0	9.1618	-0.0179
12.0	8.4	8.3693	0.0036
13.0	7.8	7.6470	0.0196
14.0	7.1	6.9879	0.0158
15.0	6.6	6.3858	0.0324

3.14 Conclusions

The author proposes that:-

a) I.D.M.T. type relay characteristics are synthesised by the rational function approximation, see 3.3.2., since it offers considerable advantages over previously used methods:- the relay equations are easily computed, give high accuracy and have a minimum storage requirement.

b) the time varying positive sequence reactance function be synthesised by an equation of the form of (3.19), derived (initially) from the values X'' , X' and X_s and the following time values:

X''	at	$t = 0$	(fault applied)
X'	at	$t = 3T''$	(the X'' term is no longer effective)
X_s	at	$t = 3T'$	(the X' term is no longer effective).

The time varying reactance function, derived from the R.M.S. current function - equations (2.15) and (2.13), and the approximation to this function, derived from equation (3.19), are shown by Fig.2.9.

c) The time varying reactance functions are included in the system analysis and that the resulting decremental currents and voltages are synthesised using equation (3.19), which has advantages similar to those listed under (a) above, see 2.5.3.

d) the results of sections (a) and (c) above are combined so that the protective system performance can be evaluated, see 3.11 and 3.12.

CHAPTER 4

PROGRAM DESCRIPTION AND CONCLUSIONS

4.1 General

The main program is, basically, comprised of three parts:

1. Symmetrical three phase load flow. (The three phase short circuit is a special case of this).

2. Asymmetrical fault analysis.

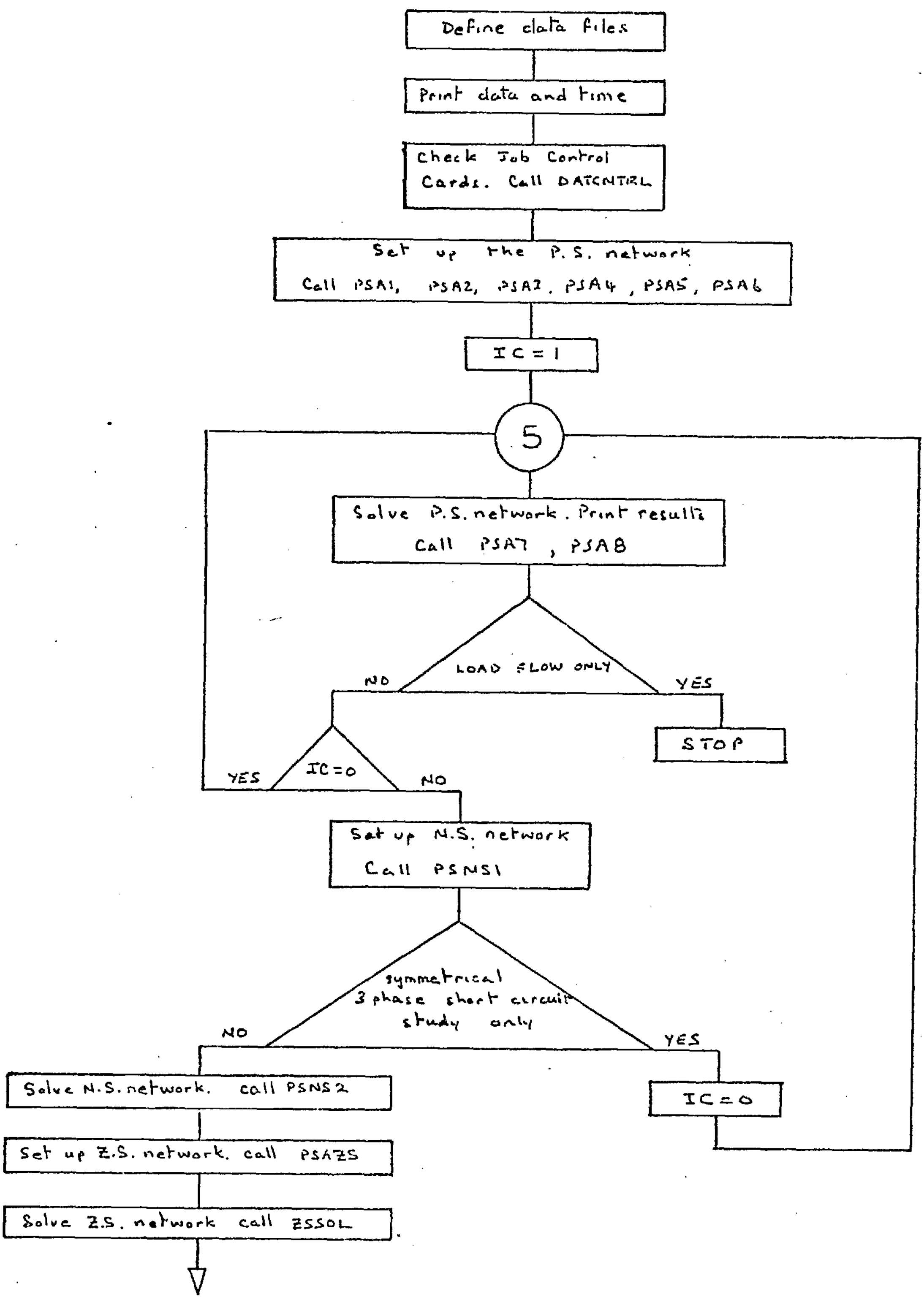
3. Protection performance.

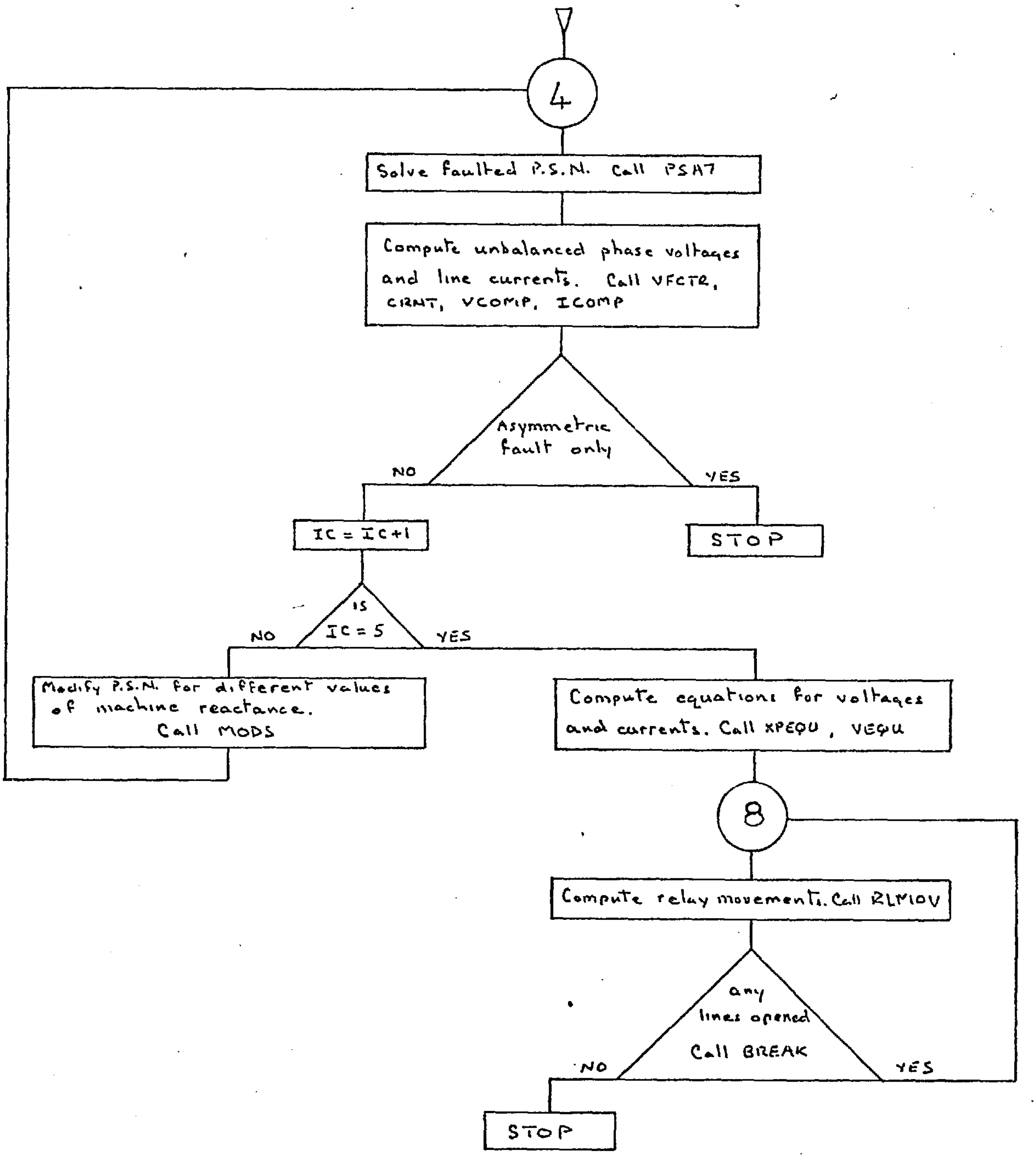
These sections do not exist as separate entities, since there are many subroutines which are common to each section. System data is read directly into a backing store file, where it is retained - if the job is a permanent job, or it is over written during a subsequent run, if the job is a temporary one. The working program therefore, obtains all required system data from the backing store.

4.2. The Controlling Routine - Subroutine PSAA

This subroutine is the master subroutine and it controls the working of the program by calling, as required, other system subroutines.

Subroutine PSAA - Flow Chart.




4
Solve faulted P.S.N. Call PSA7
Compute unbalanced phase voltages and line currents. Call VFCTR, CRNT, VCOMP, ICOMP
Asymmetric fault only
NO
YES
IC = IC+1
STOP
IS IC = 5
NO
YES
Modify P.S.N. for different values of machine reactance. Call MODS
Compute equations for voltages and currents. Call XPEQU , VEQU
8
Compute relay movements. Call RLMOV
any lines opened Call BREAK
NO
YES
STOP

4.3 Subroutine DATACNTRL

This subroutine controls the reading of the data cards and transfers the system data, as it is read, to the backing store data file. This routine also includes facilities to check that the data forms a complete and consistent set.

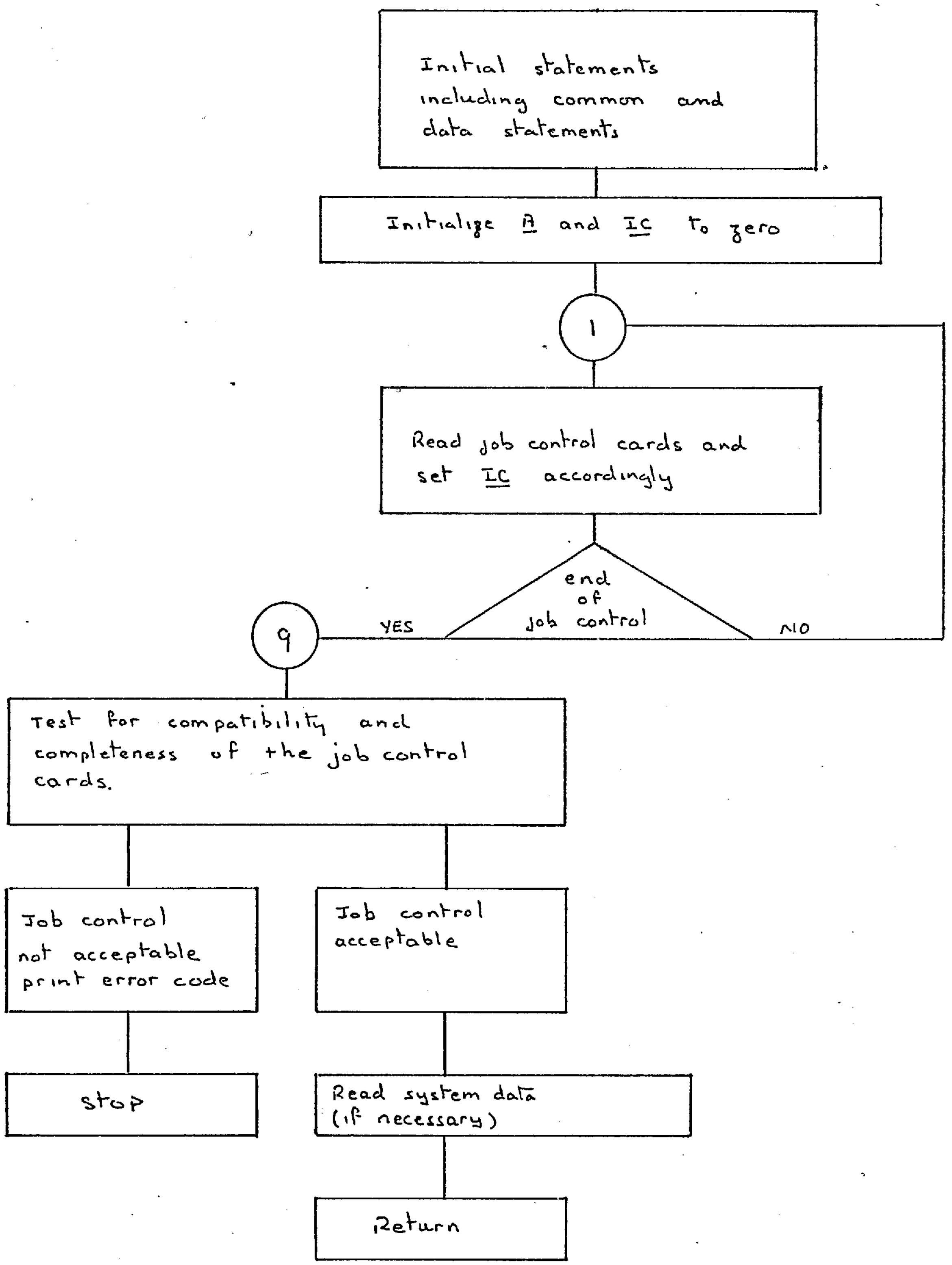
4.3.1 Subroutines PSA1 and PSA2

These subroutines read the initial system data - PSA1 reads the nodal information, and other data as specified by the job control cards. PSA2 reads the branch data. As the data is obtained, by these routines, from the backing store data file it is processed and allocated to the system data arrays.

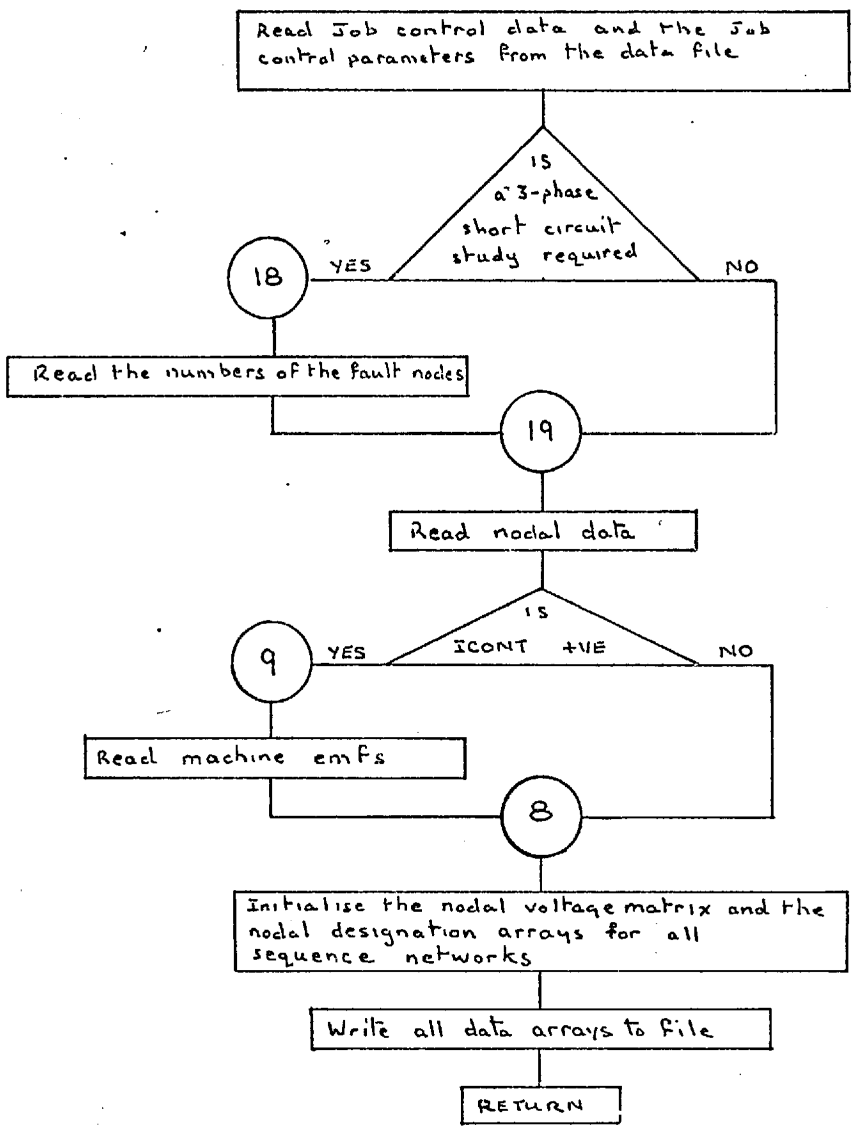
4.3.2 Subroutines PSA3 and PSA4

The branch matrix is assembled by PSA2 in a random manner which is dictated by the order that the data is presented to the card reader. Subroutines PSA3 and PSA4 work together to select elements from the branch matrix and assemble the positive sequence nodal admittance matrix.

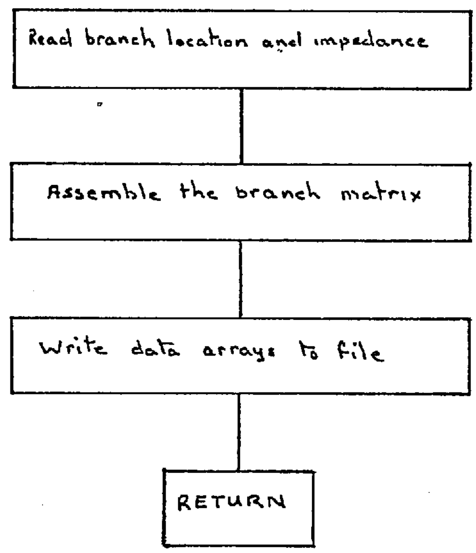
Subroutine DATACNTRL - Macro Flow Chart



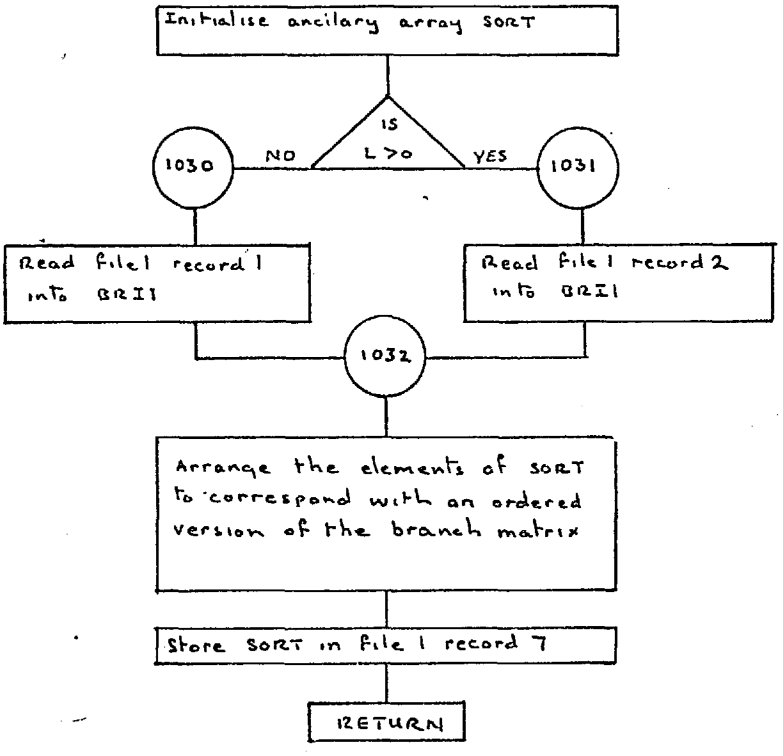
Subroutine PSA1. Flow Chart



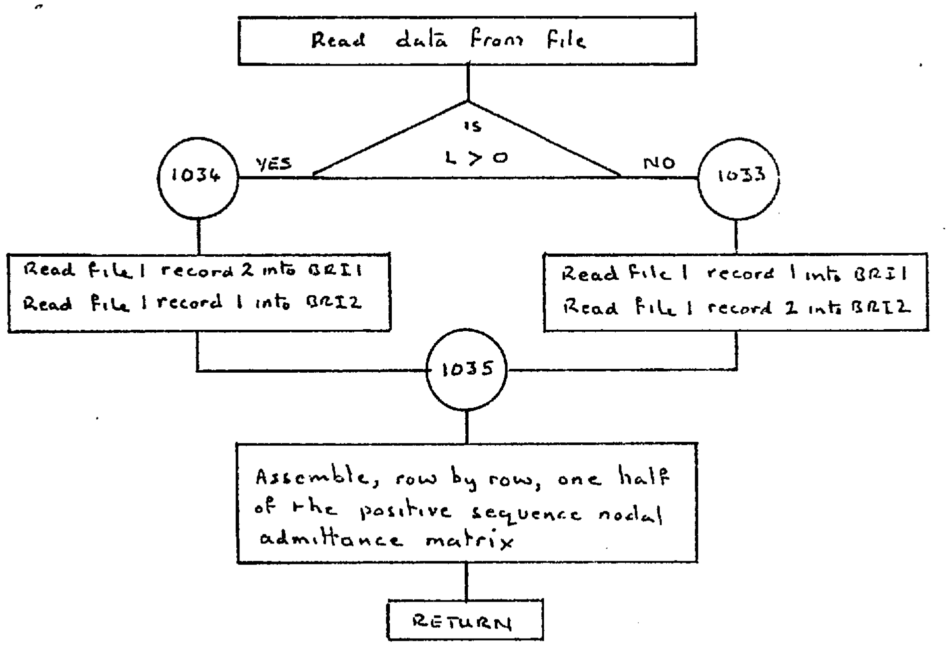
Subroutine PSA2. Flow Chart



Subroutine PSA3. Flow Chart



Subroutine PSA4. Flow Chart



4.3.3 Subroutine PSA5

Subroutine PSA5 evaluates the diagonal elements of the nodal admittance matrix. This operation is carried out on a row by row basis since each row of N.A.M. forms a record of data files 6 and 7.

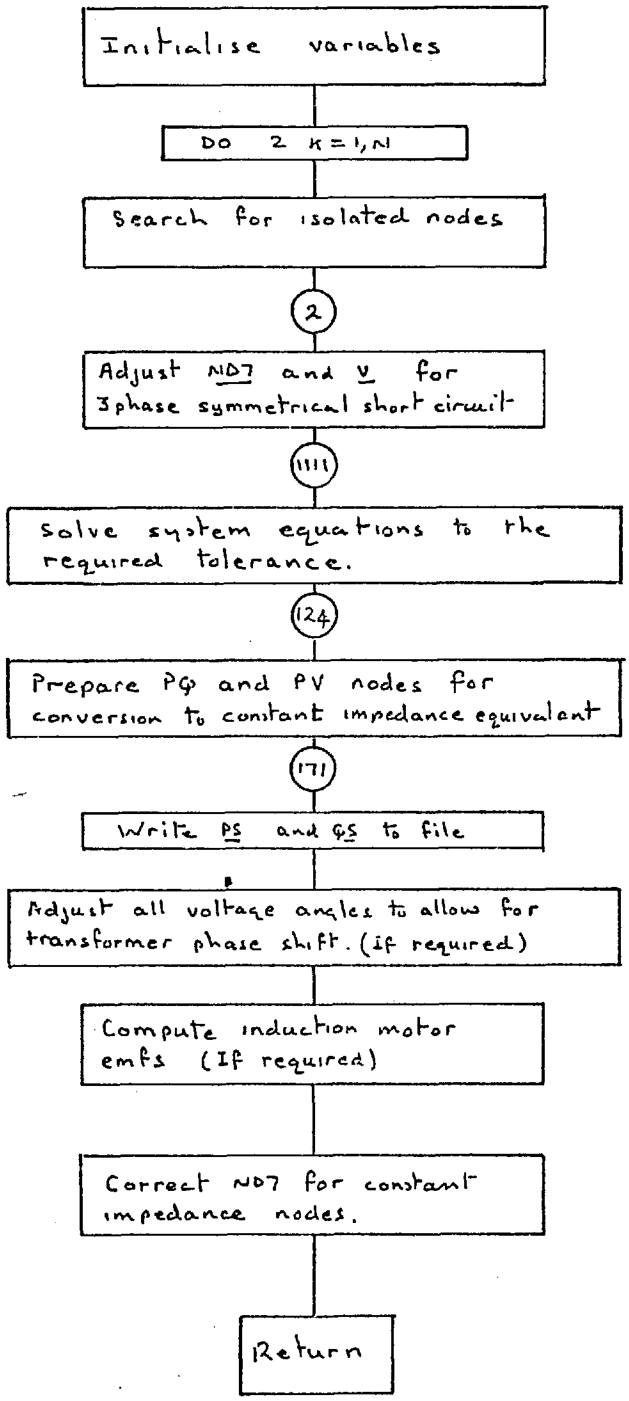
4.3.4 Subroutine PSA6

This routine modifies the completed N.A.M. to allow for transformers which are operating with an off-nominal turns ratio. It also performs a secondary function, which is the initialisation of the second branch location matrix IBUS.

4.3.5 Subroutine PSA7

The primary purpose of this subroutine is to solve the positive sequence network equations, it also performs several secondary functions - as indicated by the flow chart - such as data preparation and modification for subsequently called subroutines.

Subroutine PSA7. Flow Chart



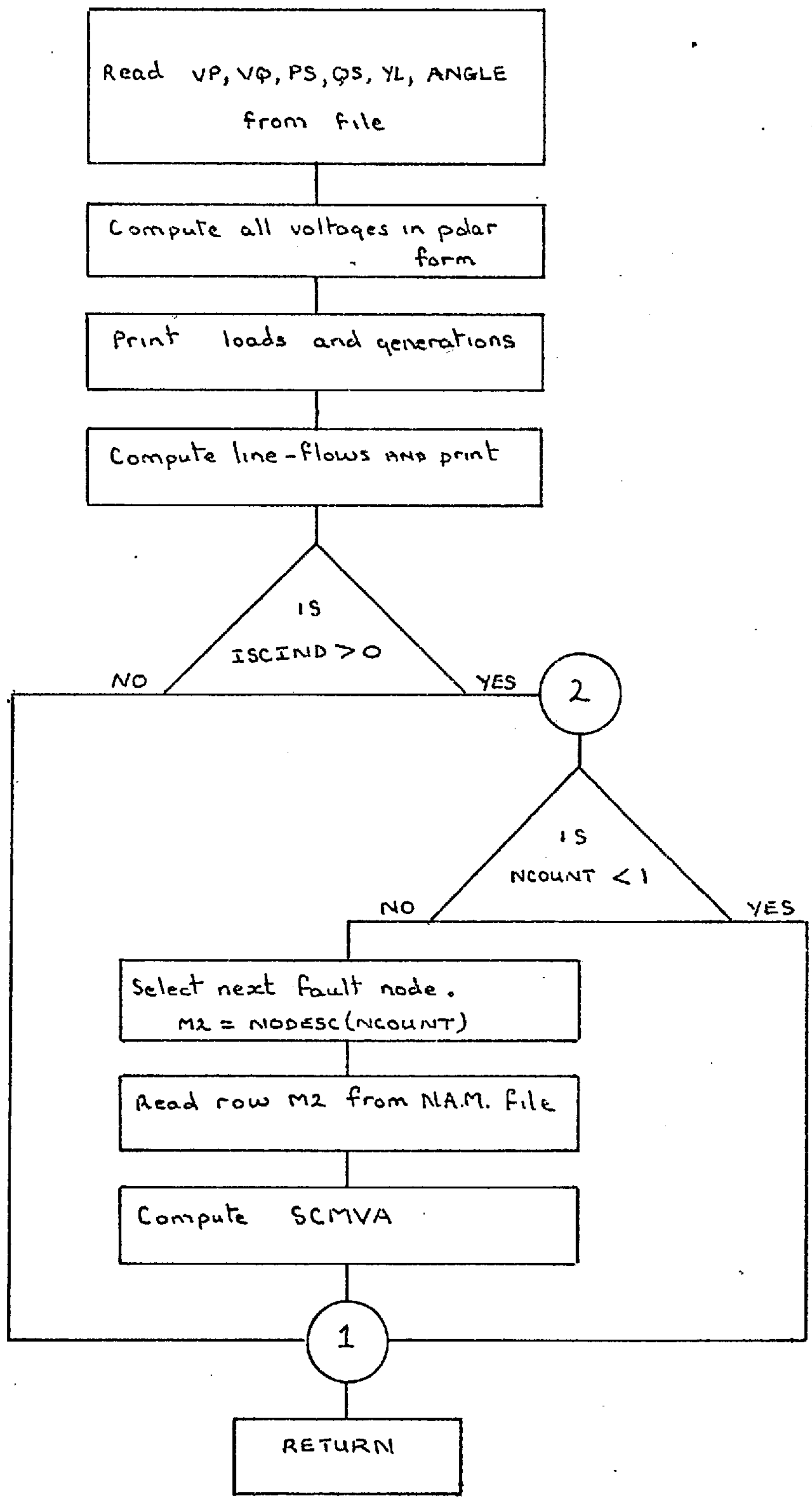
4.3.6 Subroutine PSA8

This subroutine is used to compute the line flows and to print these, and other results obtained by subroutine PSA7. It is therefore, used by two major sections of the program; the balanced load flow section and the symmetrical three phase short circuit section.

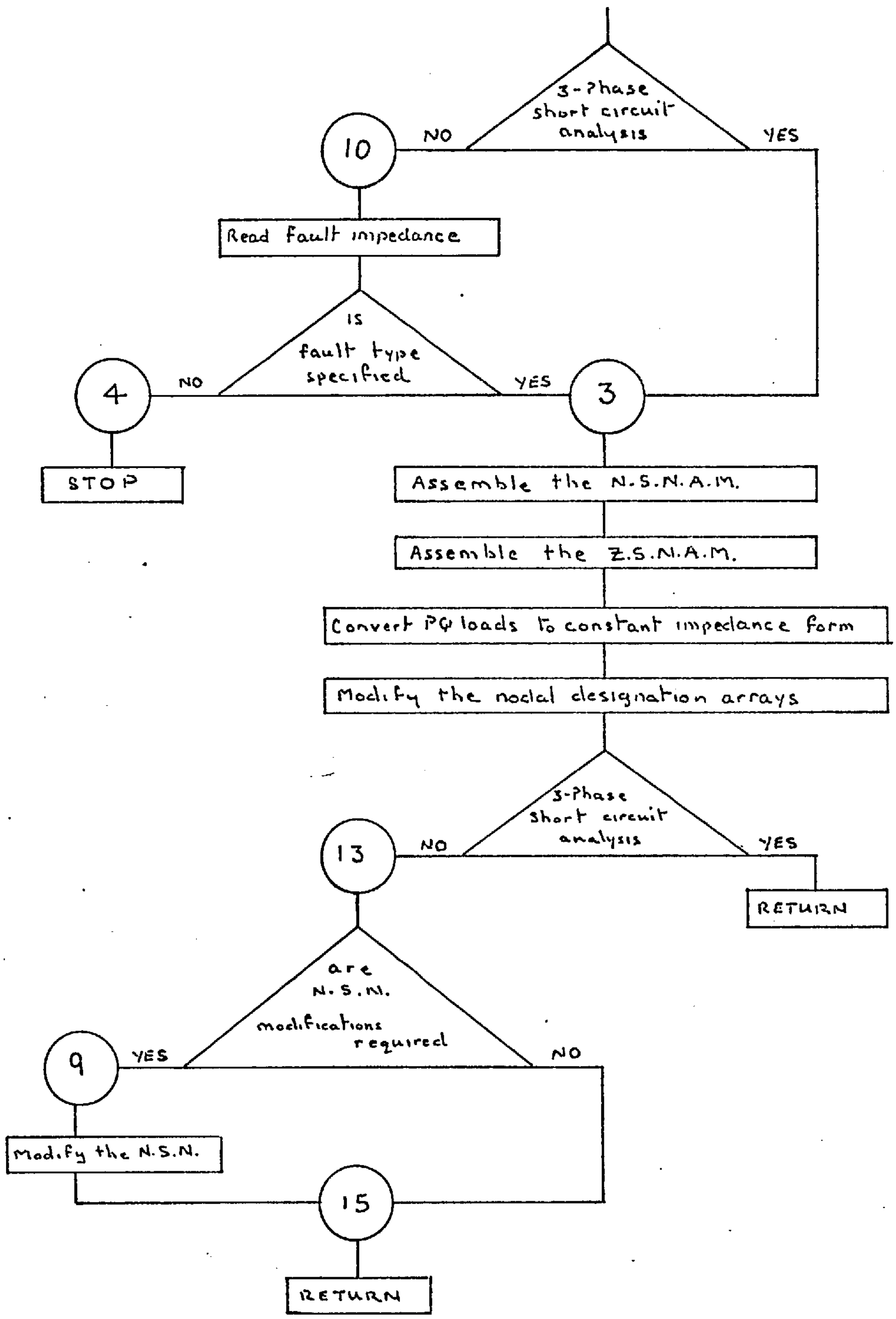
4.3.7 Subroutines PSNS1 and PSNS2

These subroutines are designed to assemble and solve the system equations that relate to the negative sequence network. PSNS1 assembles the N.S.N.A.M. from the P.S.N.A.M. data, incorporating modifications as required. It also converts loads specified as PQ loads into an equivalent constant impedance form. Subroutine PSNS2 solves the system equations using the same method (Newton-Raphson) as is used by PSA7 to solve the P.S. network equations. When a solution has been obtained, the negative sequence nodal voltages are stored in the allocated data file.

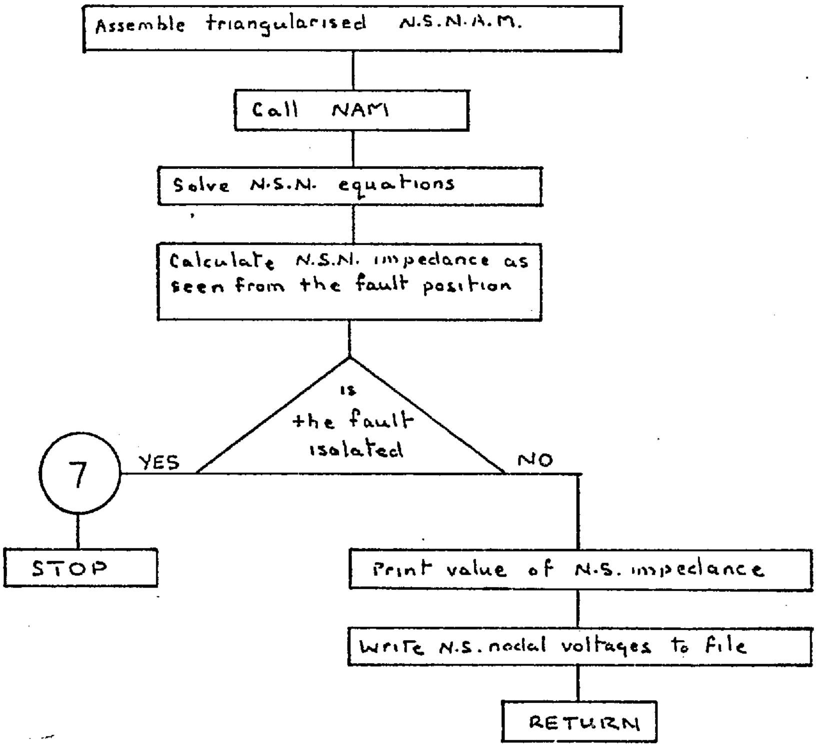
Subroutine PSA8. Flow Chart



Subroutine PSNS1. Flow Chart



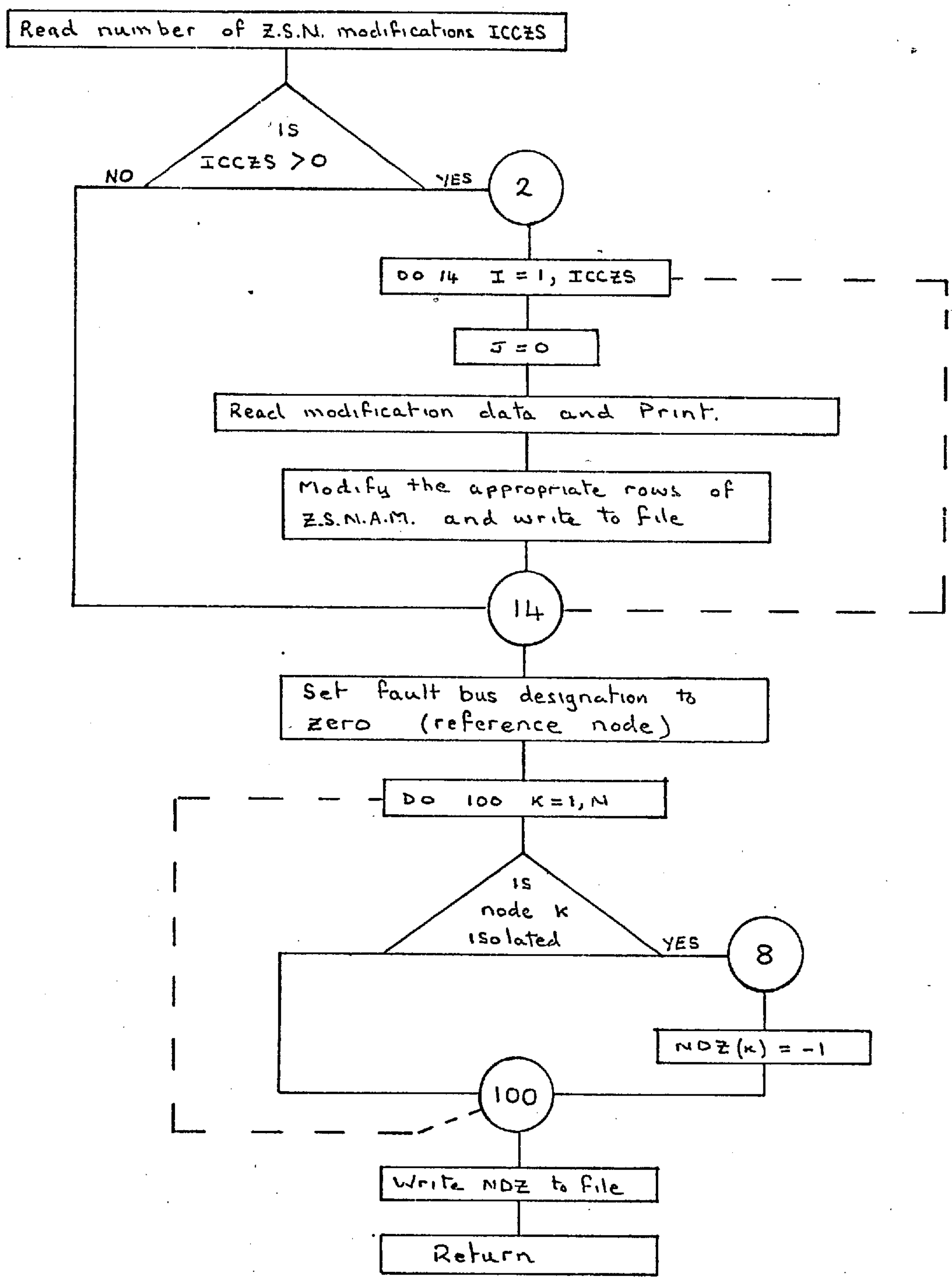
Subroutine PSNS2. Flow Chart



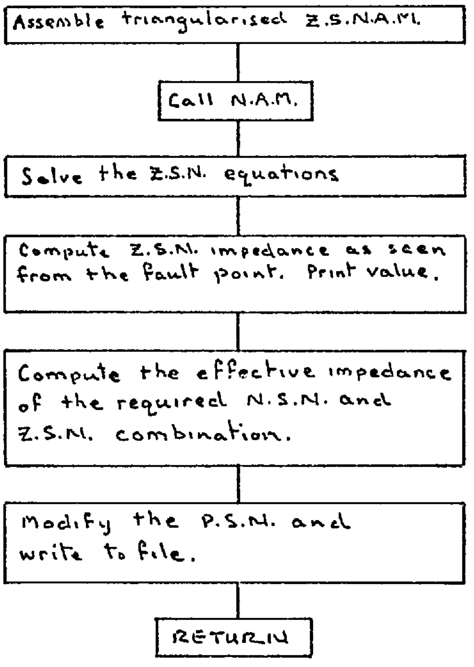
4.3.8 Subroutines PSAZS and ZSSOL

These subroutines are the zero sequence equivalents of PSNS1 and PSNS2. PSAZS completes the assembly of the Z.S.N.A.M. by including the modifications required for the transformer winding connections, as described in chapter 2. A check is made by this routine to determine if the network modifications have isolated any nodes from the remainder of the system. If an isolated node is detected, the starting value of the nodal voltage is changed to zero, and the nodal designation tag is also set equal to zero; isolated nodes are therefore, removed from the iteration procedure. The main function of ZSSOL is the solution of the Z.S.N. equations, however, it also computes the Z.S.N. impedance - as seen from the fault point - and then combines this value with the corresponding one for the N.S.N. The P.S.N. is modified as described in chapter 2.

Subroutine PSAZS. Flow Chart



Subroutine ZSSCL. Flow Chart



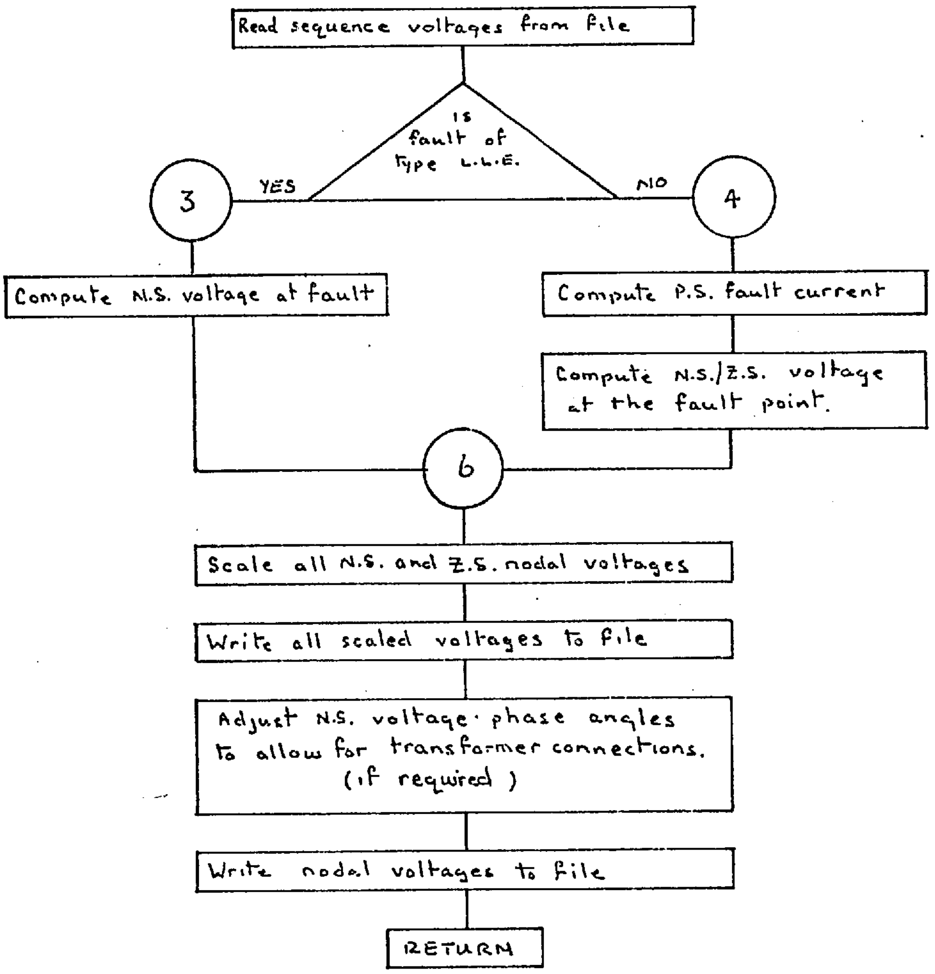
4.3.9 Subroutine VFCTR

This subroutine computes the actual negative and zero sequence voltage values at the fault point. These values are the scaling factors for the voltages which have previously been computed by PSNS2 and ZSSOL. After scaling, the sequence voltages are stored in a new set of data files.

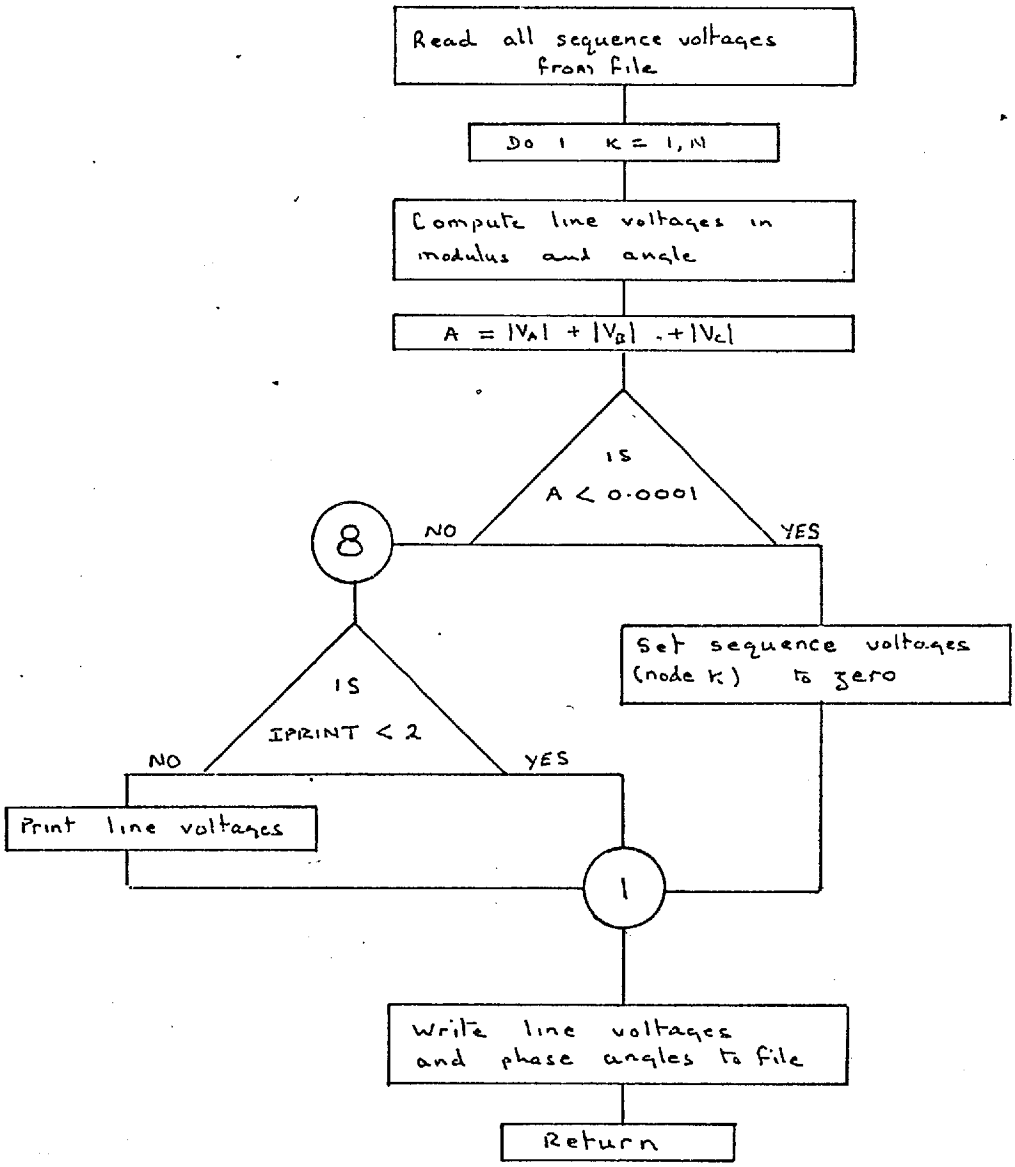
4.3.10 Subroutine VCOMP

VCOMP reads the positive, negative and zero sequence voltages from the data files, and computes the unbalanced nodal phase voltages. For an asymmetric fault analysis these results are printed, however, if a protection study is being made these results are only printed if required; the control variable is IPRINT - see User Manual, appendix 1.

Subroutine VFCTR. Flow Chart



Subroutine VCOMP. Flow Chart



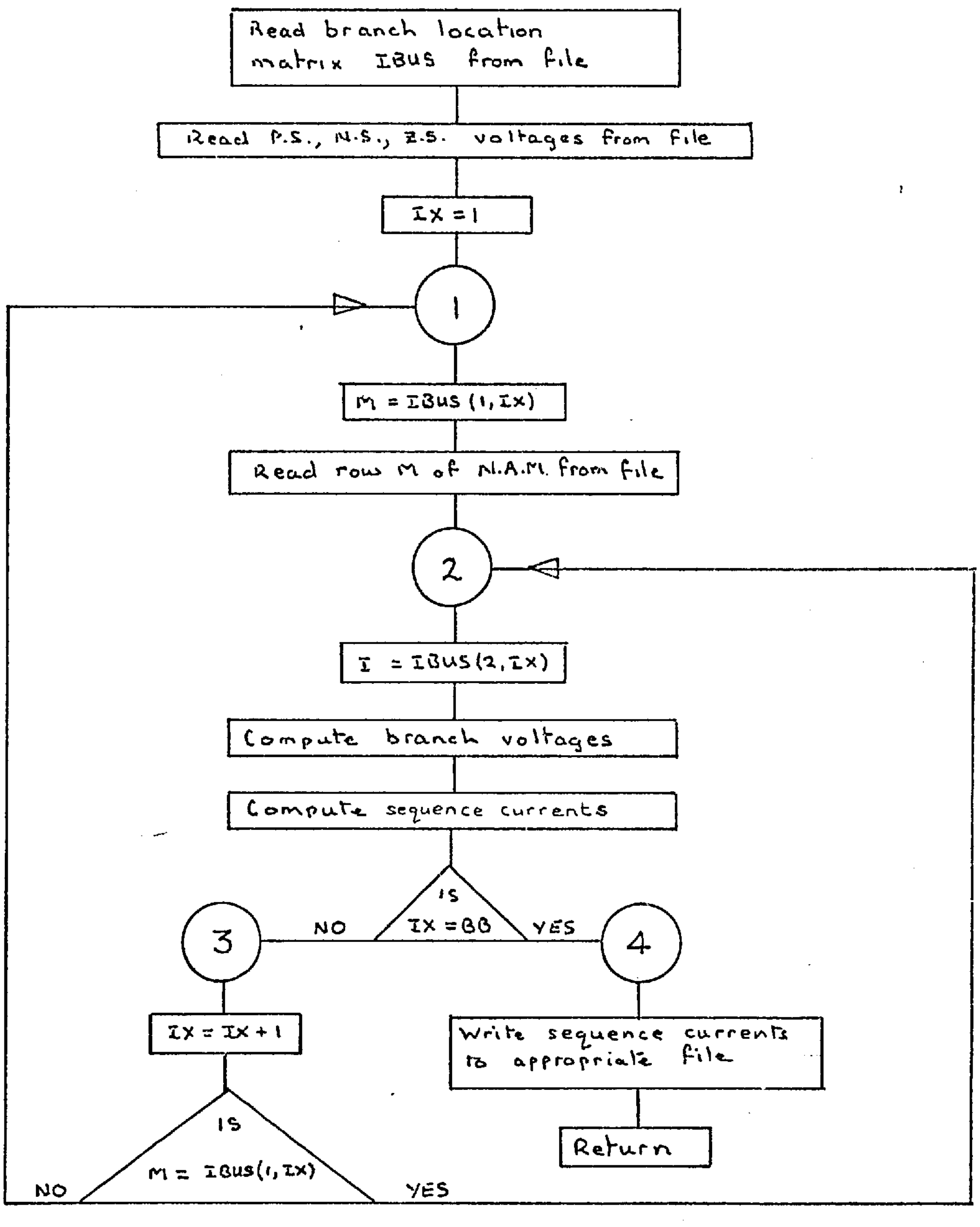
4.3.11 Subroutine CRNT

This subroutine reads the sequence voltages and the sequence network data from the data files. The branch currents for the sequence networks are then computed and stored in a data file.

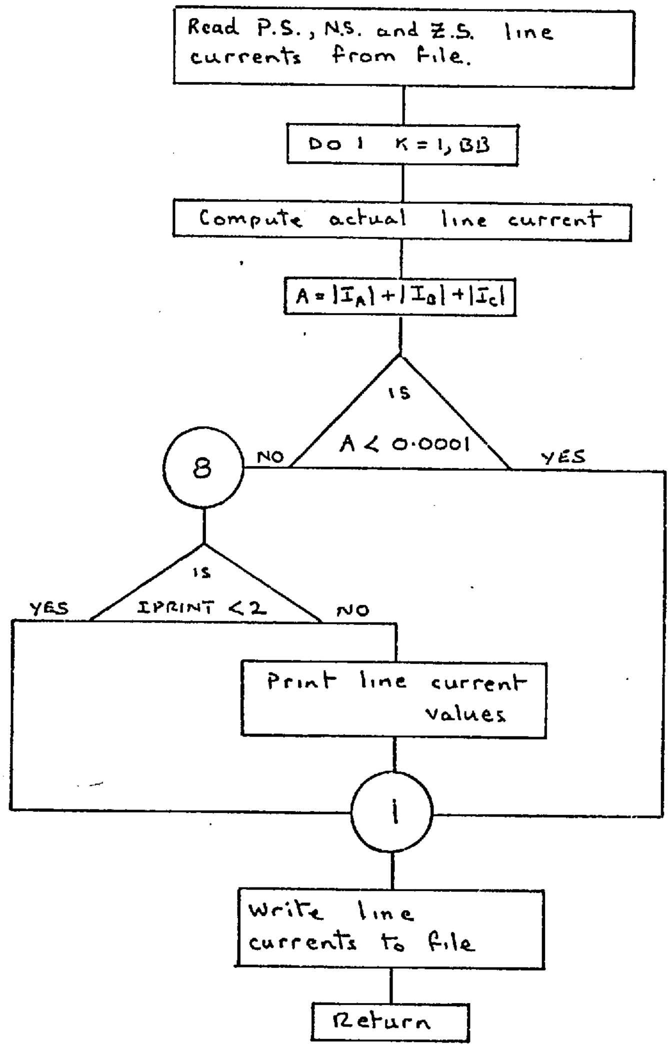
4.3.12 Subroutine ICOMP

This is a corresponding routine to VCOMP, it reads the sequence currents from the data files and computes the actual system line currents. The reason for having two routines doing such a similar job is that the data files for the voltages are of a different length to the data files which hold the sequence currents, and the arrays in the subroutines have to be dimensioned so as to be equal in length to the data files.

Subroutine CRNT. Flow Chart



Subroutine ICOMP. Flow Chart



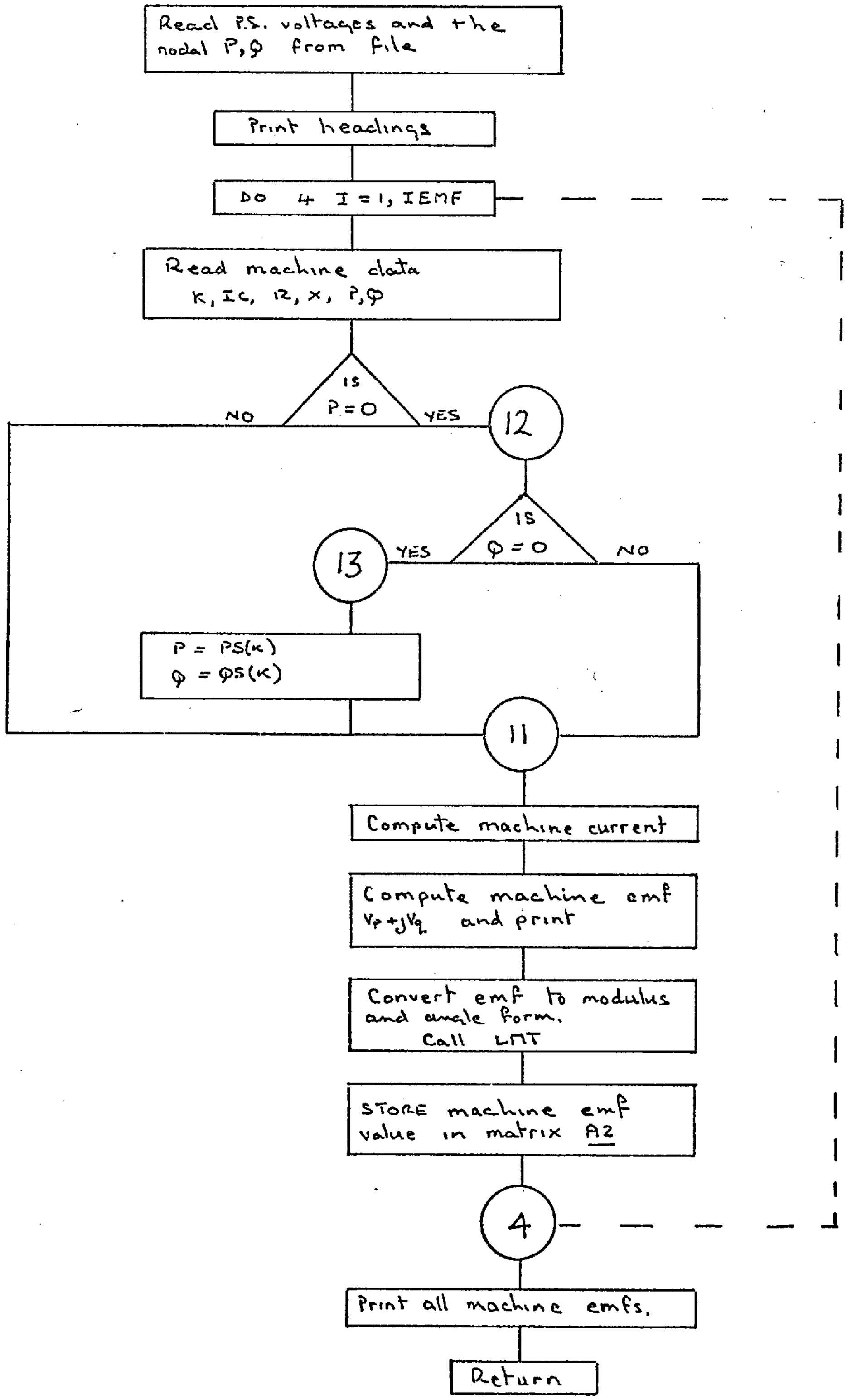
4.4 Subroutine EMF

This routine is used to compute the value of the emf acting behind the machine impedance. To accomplish this the results of the balanced load flow analysis are used, together with some additional data - see User Manual, appendix 2.

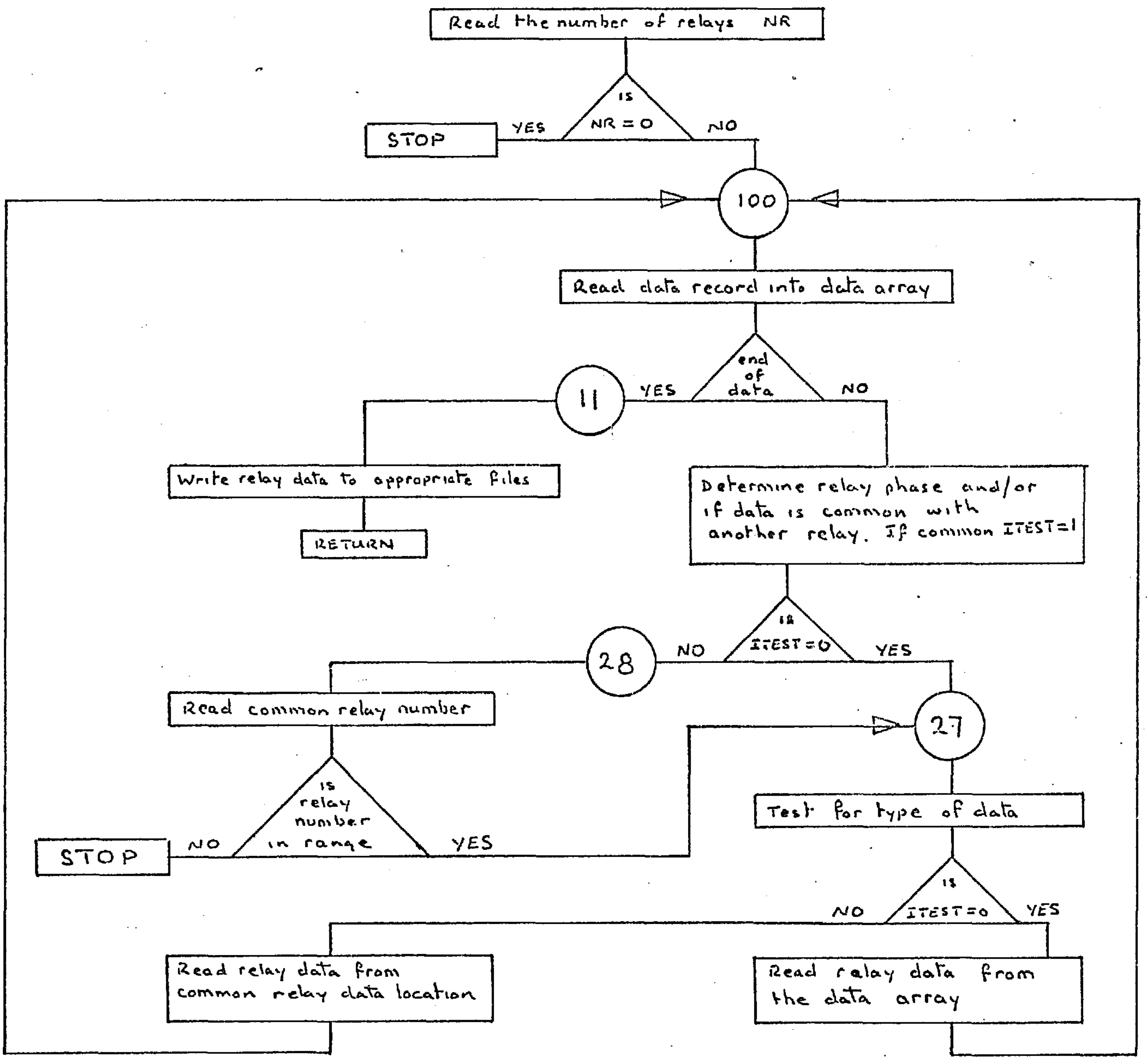
4.4.1 Subroutine RLDAT

Subroutine RLDAT is used to read the relay data from the main data file, and to distribute this data to the various relay data files. In order to minimise the amount of relay data it is possible to use the data which has been supplied under a given relay number for other relays as well. Subroutine RLDAT will find and copy such data into the required locations. This facility, and the procedure for using it are fully described in the User Manual.

Subroutine EMF. Flow Chart



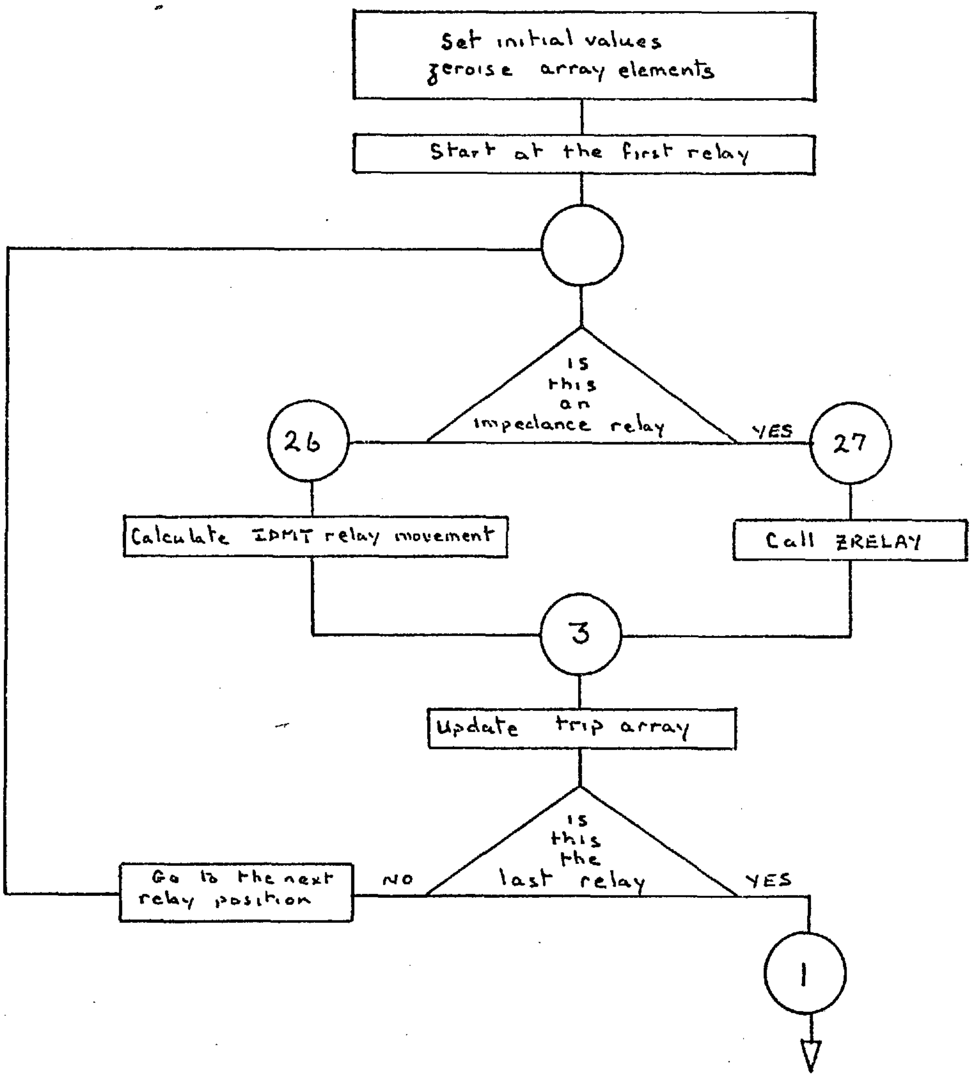
Subroutine RLDAT   Flow Chart

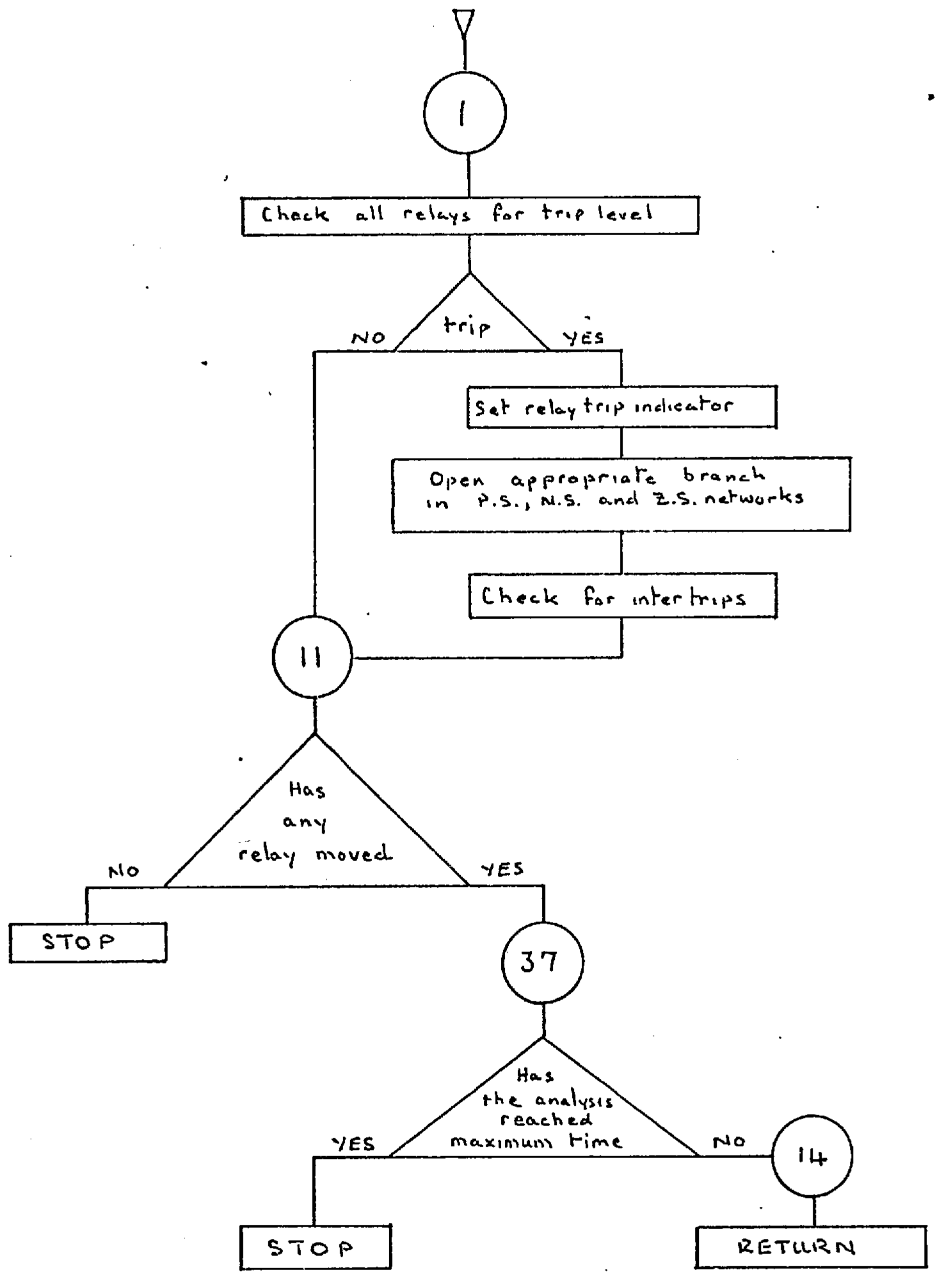


4.4.2 Subroutines RLMOV, ZRELAY and OPEN

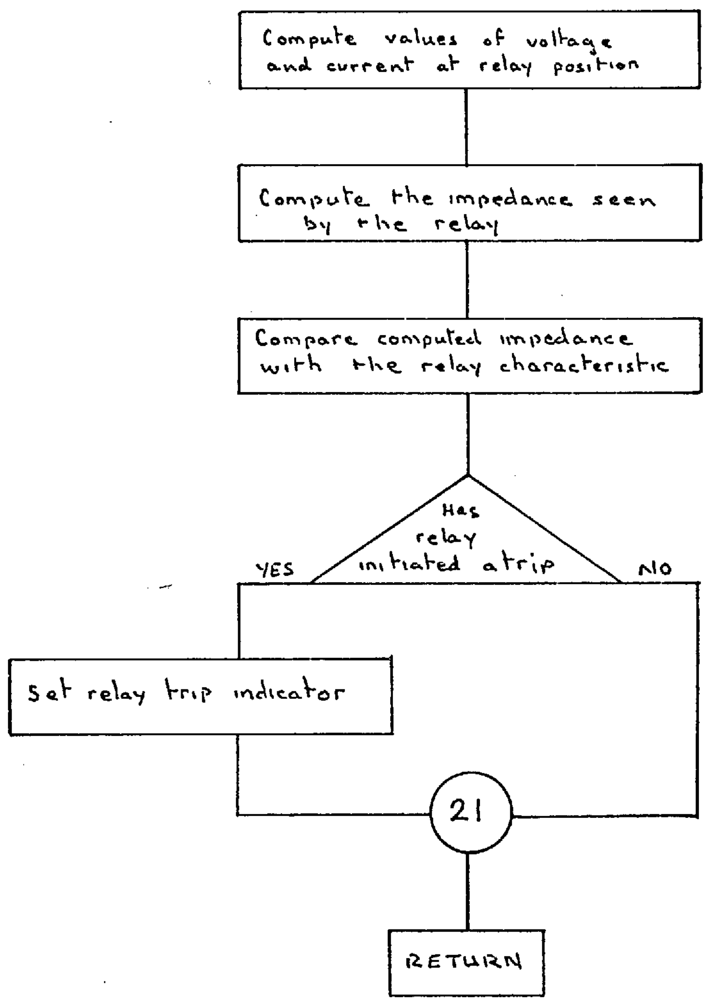
Subroutine RLMOV determines the movement of each relay, and controls the subsequent program action following the evaluation of the relay movements for any given time interval. This is achieved with the assistance of two auxiliary subroutines ZRELAY and OPEN. RLMOV computes the action of I.D.M.T. type relays and ZRELAY is used to compute the action of impedance relays. Subroutine OPEN modifies the sequence networks following circuit breaker action.

Subroutine RLMOV Flow Chart

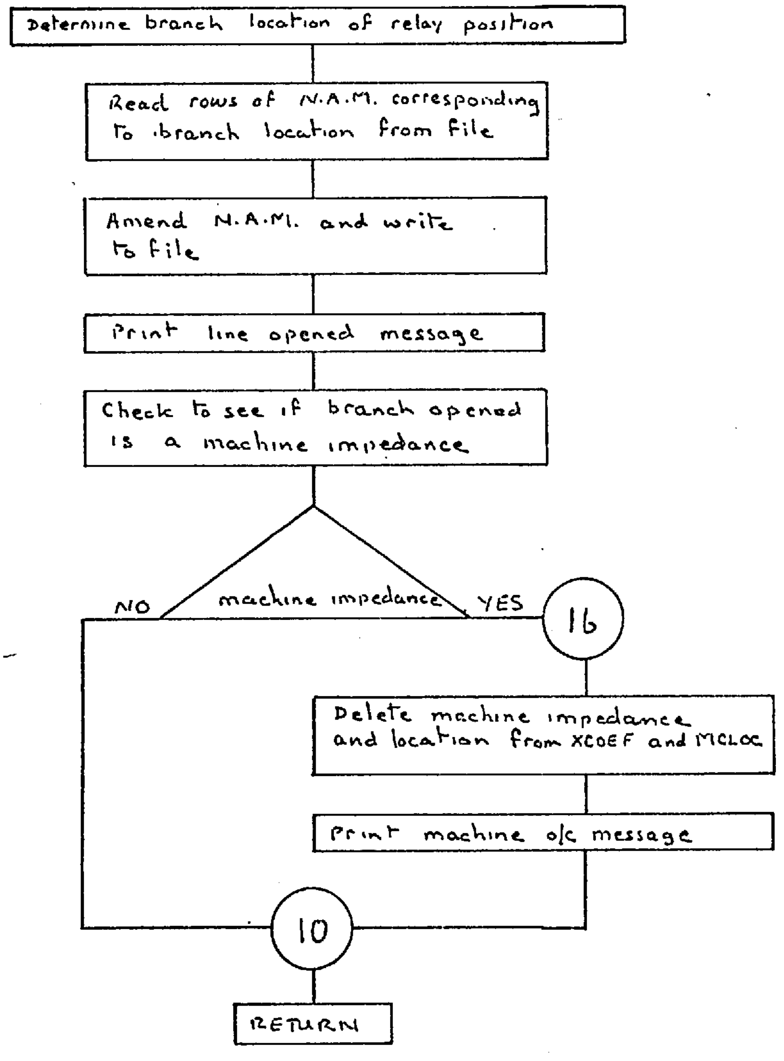



1
Check all relays for trip level
trip
NO
YES
Set relay trip indicator
Open appropriate branch in P.S., N.S. and Z.S. networks
Check for intertrips
11
Has any relay moved
NO
YES
STOP
37
Has the analysis reached maximum time
YES
NO
14
STOP
RETURN

Subroutine ZRELAY Flow Chart



Subroutine OPEN Flow Chart



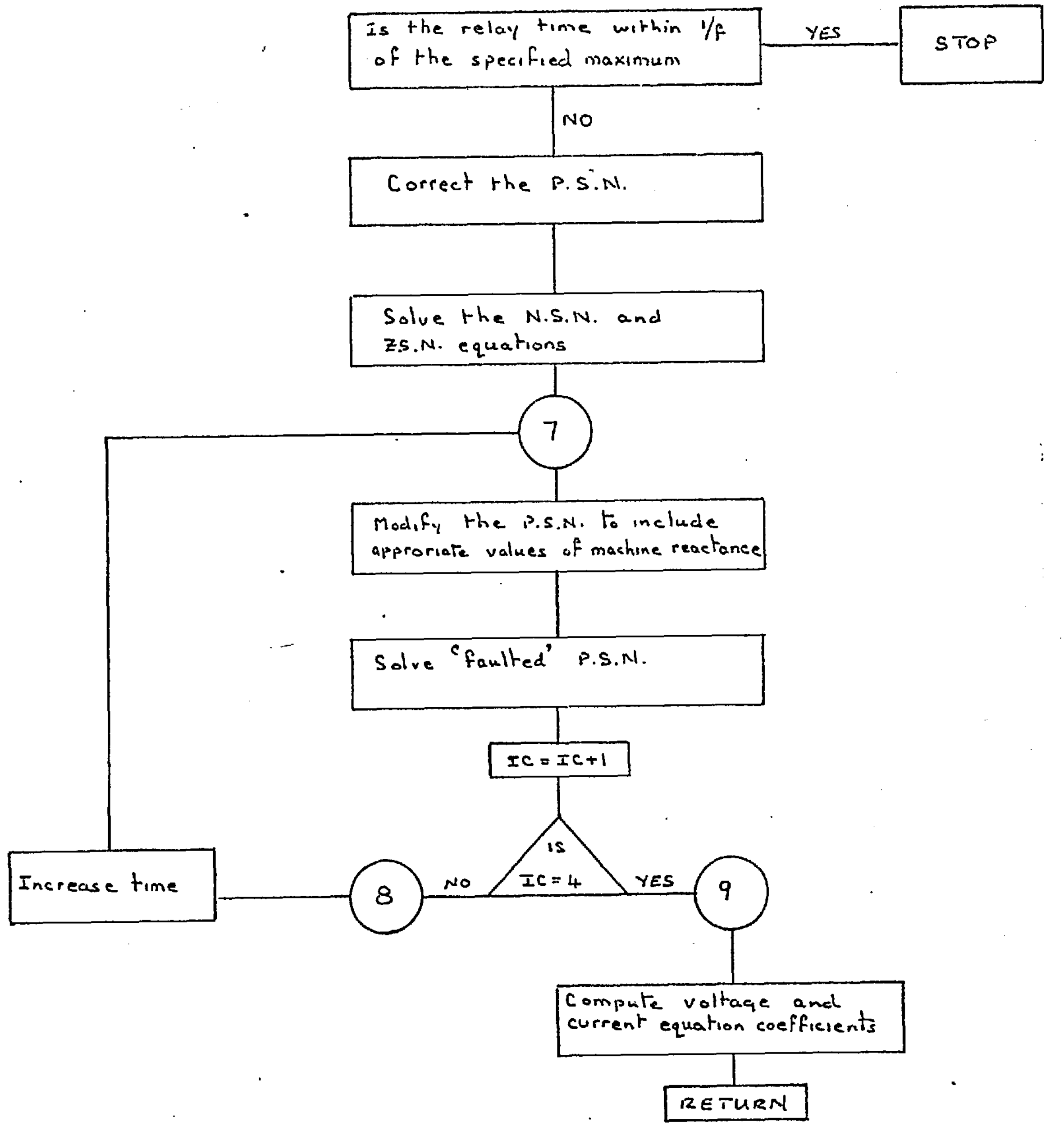
4.4.3 Subroutine BREAK

Subroutine BREAK acts as a minor master routine. It is called after subroutine OPEN has modified the sequence networks to allow for circuit breaker action. BREAK calls PSNS2 and ZSSOL to solve the modified negative and zero sequence networks. The positive sequence network equations are then altered so as to include the effective negative and zero sequence impedances, and also to include the appropriate values of the machine reactances. This procedure is repeated for three different values of time so that the decremental equations for the system voltages and currents can be computed.

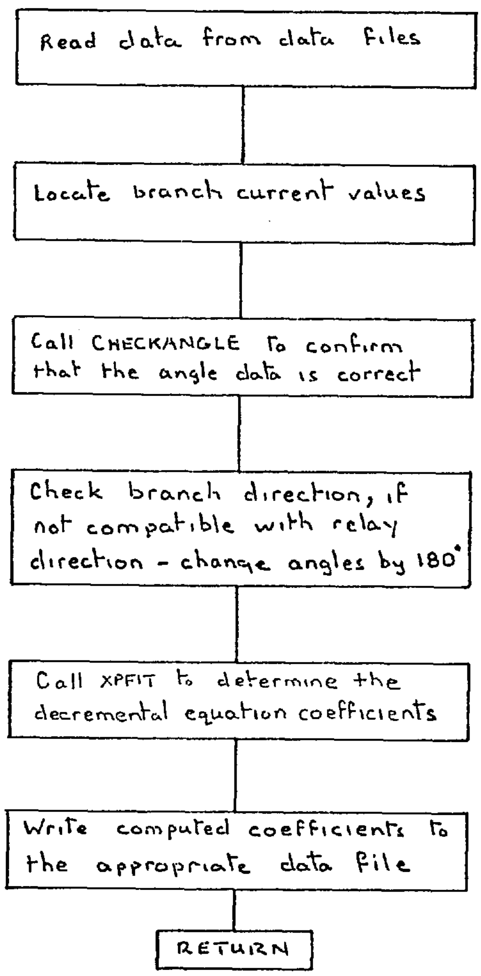
4.4.4 Subroutines XPEQU, XPFIT, VEQU and CALC

This collection of subroutines is used to determine the coefficients of the decremental equations. XPEQU is the controlling routine for the branch current equations and VEQU is the controlling routine for the nodal voltage equations. XPFIT is the subroutine which evaluates the coefficients and calls the subsidiary routine CALC which checks the computed function to see if it generates any negative values within the specified time range. If negative values are detected a warning message is printed together with the values of the function.

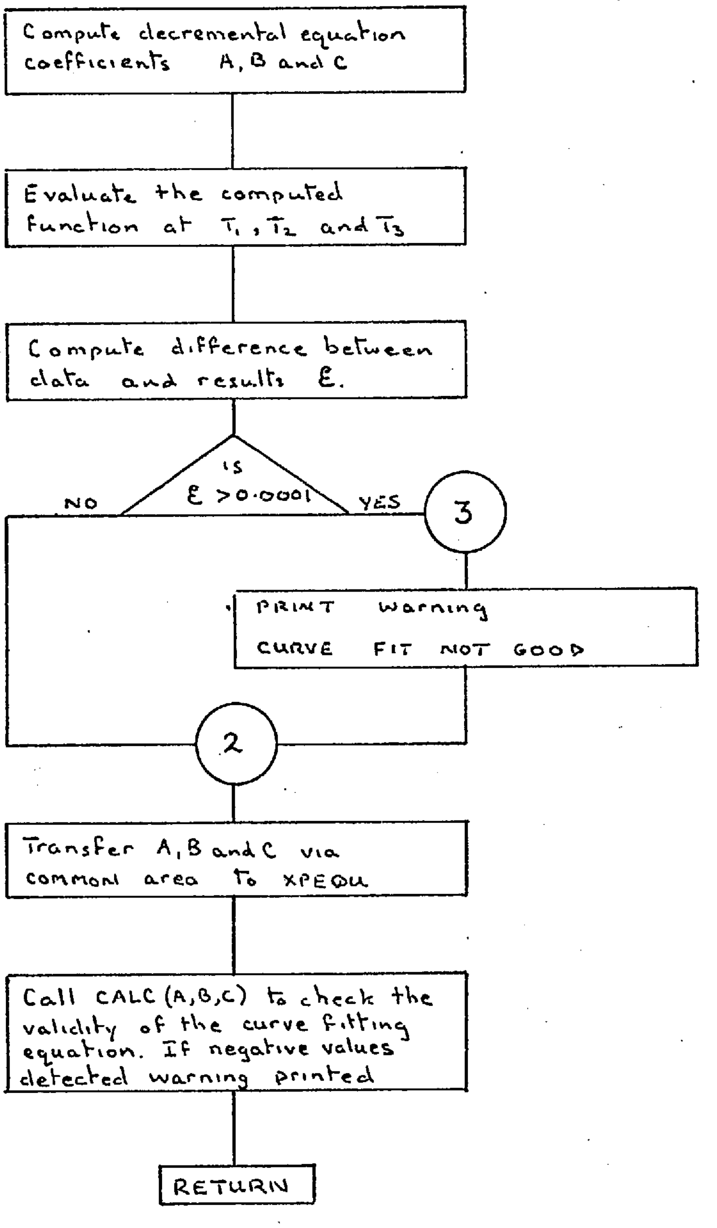
Subroutine BREAK Flow CHART



Subroutine XPEQU Flow Chart



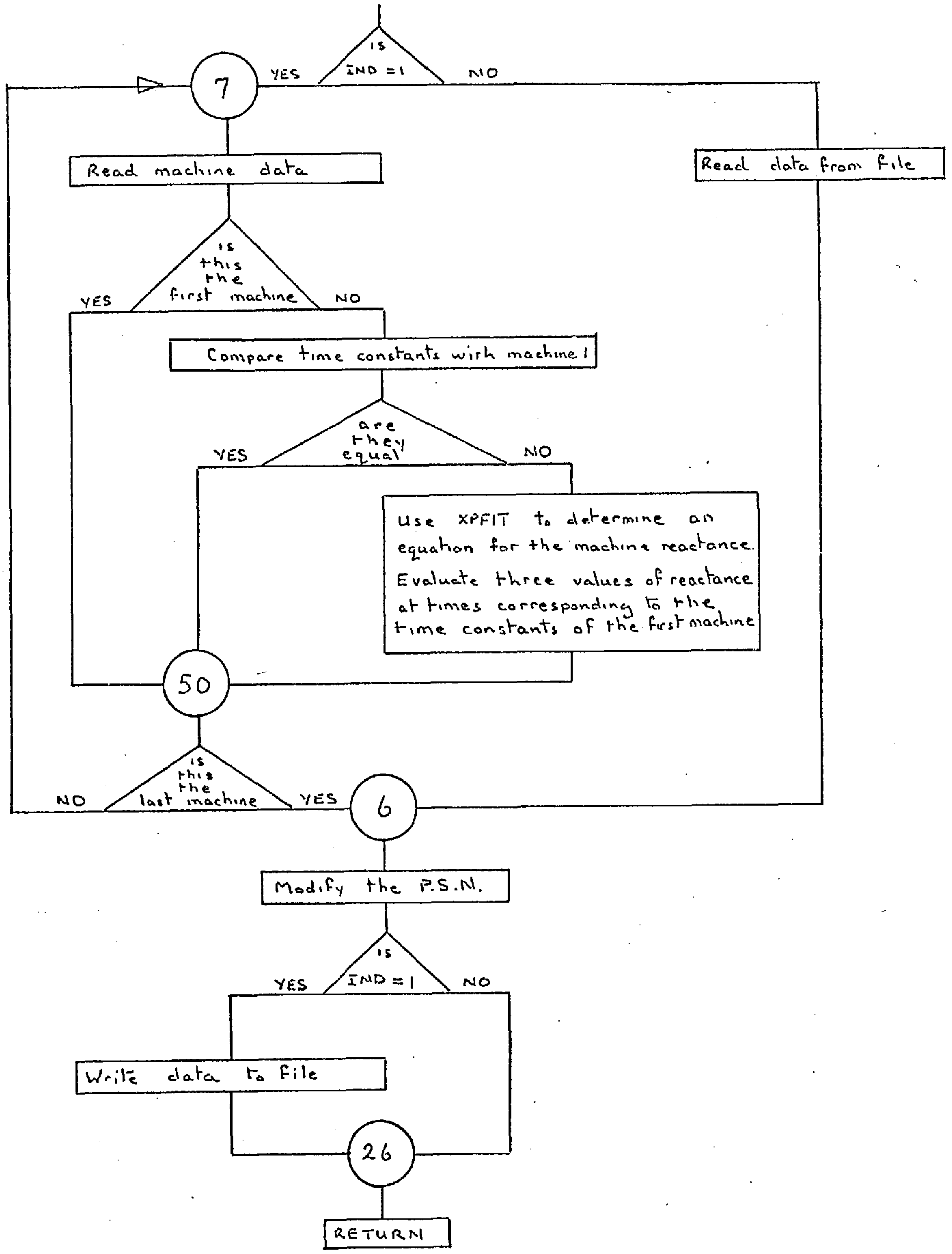
Subroutine XPFIT Flow Chart



4.4.5 Subroutine MODS

This routine is used to modify the positive sequence network so that the three different values of machine reactance are incorporated (sequentially). Initially these values will correspond to X" , X' and X_s for the first machine, at least. For all other machines the values of reactance used will depend on the time constants of each machine.

Subroutine MODS Flow Chart



4.5 The Data Storage and Handling System

As described in section 4.3 DATACNTRL reads the initial system data 'en bloc' into the main data file, an auxiliary routine CREAD is used for this purpose. This data file is accessed by various subroutines in the program (via an auxiliary routine FREAD) which use and/or process this data before reallocating it to a set of working data files. This set of data files corresponds, approximately, with the various types of data used in the program. For example, File 15 is used to store voltage information, as follows:

File 15.	Records	Data
	1 & 2	P.S. Voltages.
	3 & 4	Iteration starting values.
	5 & 6	N.S. Voltages (scaled).
	7 & 8	Z.S. Voltages (scaled).
	9 & 10	N.S. Voltages (unscaled).
	11 & 12	Z.S. Voltages (unscaled).
	16 to 33	Moduli and angles of the unbalanced nodal phase voltages.
	41 to 44 and 51 to 54	Iteration values accessed by the acceleration scheme.
	56 & 57	Results of the acceleration scheme.

Thus subroutine PSA7 will avail itself of the nodal voltage starting values in records 3 and 4, and when the solution values have been determined they will be written into records 1 and 2, where they are available to all other subroutines as required.

The above procedure is necessary since the size of the program dictates that the overlay system is extensively used. This aspect of the program is described in detail in the User Manual, Section 6.48.

4.6 Conclusions

1. This development enables a more exact determination than has hitherto been possible, of the fault currents in multimachine industrial distribution systems and thus the evaluation of correct circuit breaker ratings. It also enables the performance of the protection applied to such systems to be evaluated.

2. The computer program which has been developed through this work differs from those described by previous workers (17, 19, 20, 21) in three ways:

a) The functions developed by the author to represent the relay characteristics are continuous and accurate to better than one per cent over the whole of the characteristic range. Alderton and Peralta (20) use two equations which give maximum and minimum operating time values i.e. the characteristic is represented by a band of time values. Graham and Watson (17) use tabulated characteristics, an approach of obviously limited accuracy. Both Albrect (19) and Begian (21) use logarithmic functions, which in Albrects program require 20 constants for each relay, and in Begians program require 7 constants plus auxilliary functions and operations.

b) The authors program evaluates the values of the decremental line currents and nodal voltages in the faulted system at any required time. Graham and Watson (17), Begian (21) and Albrect (19) base all relay calculations on system currents and voltages that have been evaluated at one instant of time only.

c) Alderton and Peralta (20) have designed a relay operation checking program for transmission systems, and therefore, unlike the authors program does not require a provision for the representation of induction motor loads. However, it is based on a transient stability analysis and can therefore include the effect of synchronous machines within the system. Graham and Watson, and Albrect do not allow for the effect of induction motor loads in their programs, which have been designed to determine the initial relay settings in new systems.

3. Program Uses

This program may be used as a tool by design engineers to ascertain system fault levels which include the contribution made by motor loads. It will, thus, be possible to determine the correct ratings of circuit breakers. The program can also be used to check the operation of a proposed protective system, thereby allowing modifications to be made, if required, before installation.

An operations engineer will be able to use this program to confirm that the existing relay settings will not lead to unplanned circuit breaker operation, or non-operation, when maintenance, with the associated line outages, is taking place. It will also be a useful aid in checking the overall protection performance of a system as that system is extended and modified.

4. Use of Appendices

Two appendices have been written to enable future workers to fully understand and use the program developed through this work.

4.1 User Manual (Appendix 1)

The User Manual may be regarded as having three sections, the first of which comprises paragraphs 5.1 to 5.3 inclusive. This section specifies the minimum computer configuration necessary to operate the program, and also describes the physical size limitations of the networks which may be analysed.

The second section, paragraphs 5.4 and 5.5, specifies the order, type and format of all possible data cards. This section should be used to compile, in appropriate form, the data for any given problem. The text of this section indicates when individual items of data and/or complete data cards may be omitted.

Section 3, paragraph 5.6, contains a specimen problem with a complete annotated listing of the data cards. This example may be used to clarify the text of section 2. Finally, the results from this problem are given and may be used for comparison purposes when the program is tested after being transferred to another computer.

4.2 Detailed Program Description (Appendix 2)

Appendix 2 is a detailed description of the contents and working of the complete program. This appendix will enable users who wish to extend or amend the program to understand the operation of each subroutine and the inter-relationships between subroutines. These inter-relationships form a complex system since the operation of the program relies heavily on the use of a direct access backing store which is subdivided into a series of files. These files are described and listed in section 6.47. Also, the program uses four levels of overlay, described in 6.48, future workers must ensure that the overlay integrity is not violated (a return cannot be made to an overlayed subroutine which has been overwritten).

This appendix is completed with a program listing, which may be used to clarify any problems arising from data handling and/or error messages which a new user may have difficulty in interpreting.

5. Future Work

It seems likely that at least one Electricity Board will use the program for protection operation checking, and as a consequence the program could be provided with a facility to circumvent two of the load flows and the curve fitting routines. Relay operation being determined solely from the results of a single load flow, all machines, where included, using a value of reactance equal to the transient reactance.

The amount of I.D.M.T. relay data could be significantly reduced by allowing each relay to access an appropriate characteristic in a library file; as only three or four different relay characteristics are likely to be required. At the present time the program requires that each relay be supplied with its own characteristic.

The immediate access core store requirements could be significantly reduced by adopting sparsity programming techniques in subroutines PSA7, PSNS2 and ZSSOL.

At the present time this program can model I.D.M.T. and Impedance relays; a further provision for 'Mho' type relays will obviously be useful.

The introduction of a variable step length in the integration routines which are used to calculate the relay movements would significantly reduce the computation time required for this process.

References

1. Huening. W.C. Time Variation of Industrial Short Circuit Currents and Induction Motor Contribution. Trans. A.I.E.E. May 1955. Vol. 14 Part 2, 90 - 101.

2. Cooper. C.B. The Transient Performance of Industrial Power Systems. A.E.I. Engineering, Sept/Oct. 1966 250 - 256.

3. Cooper. C.B., MacLean. D.M., Williams. D.G. Application of Test Results to the Calculation of Short Circuit Levels in Large Industrial Systems with Connected Induction Motor Loads. Proc. I.E.E. Nov. 1969. Vol. 116, No.11, 1900 - 1906.

4. Douglas. A.M., and Khan. I.U. The Effect of Induction Motors on Fault Levels and Switchgear Requirements in Distribution Systems. Presented at I.E.E. Power Division on main power distribution in industry, London 1966.

5. Wagner. W.P. Short Circuit Contribution of Large Induction Motors. Proc. I.E.E. June 1969. Vol. 116, No.6, 985 - 990.

6. Kalsi. S.S., Stephen. D.D., Adkins. B. Calculation of System Fault Currents due to Induction Motors. Proc. I.E.E. Jan. 1971. Vol. 118, No.1, 201 - 215.

7. Kalsi. S.S., and Adkins. B. Transient Stability of Power Systems Containing Both Synchronous and Induction Machines.
Proc. I.E.E. Oct. 1971. Vol. 118, No.10, 1467 - 1474.

8. Ward. J.B., and Hale. H.W. Digital Computer Solution of Power Flow Problems.
Trans. A.I.E.E. June 1956. Vol. 75, No.111, 398 - 404.

9. Fortescue. C.L. The Method of Symmetrical Components Applied to The Solution of Polyphase Networks.
Trans. A.I.E.E. Vol. 37, 1918. 1627 - 1640.

10. Say. M.G. Performance and Design of Alternating Current Machines.
Pitman 1958, 414 - 423.

11. Adkins. B., The General Theory of Electrical Machines.
Chapman and Hall 1964, 188 - 192.

12. Assefi. S. M.Sc. Thesis, Imperial College of Science and Technology, 1971.

13. Electrical Transmission and Distribution Reference Book.
Westinghouse Electric Corporation. 1950.

14 Freris. L.L. and Sasson. A.M. Investigation of the Load Flow Problem.
Proc. I.E.E. Oct. 1968. Vol. 115, No.10, 1459 - 1470.

15. B.S.S. 142. 1966.

16. Protective Relays Applications Guide. The English Electric Co. Ltd.

17. Graham. R.A., and Watson. R. Computation of Relay Performance in Electrical Distribution Systems. (Conference on) the application of computers to power system protection and metering, held in Bournemouth, May 1970. 39 - 49.

18. Heiber. J.E., Empirical Equations of Overcurrent Relay Curves for Computer Applications. I.E.E.E. Conference Paper 31-CP-65-91 1965.

19. Albrect. R.E. et al. Digital Computer Protective Device Co-ordination Program 1. Trans. I.E.E.E., April 1964. Vol. 83 No.4, 402 - 410.

20. Alderton. J.R., and Peralta. R.V. A Program to Check H.V. ac System Protection Grading. (Conference on) the application of computers to power system protection and metering, held in Bournemouth, May 1970. 218 - 227.

21. Begian. S.S. A Computer Approach to Setting Overcurrent Relays in a Network. Electrical World (U.S.A.), Vol.159, May 1963. 447 - 457.

22. Application of Protective Relays and Devices to Distribution Circuits.
I.E.E.E. Committee Report. Oct. 1964. P.A.S. 83(10), 1034 - 1042.

23. Radke. G.E. A Method for Calculating Time Overcurrent Relay Settings by Digital Computer.
Trans. I.E.E.E. 1963. Special Supplement, 189 - 204.

24. Discussion on Reference 23. Trans. I.E.E.E. March 1966. P.A.S. 85(3), 303 - 307.

25. Scheid. F. Numerical Analysis.
Schaum's Outline Series. McGraw Hill, 1968.

26. Handscombe. D.C. Methods of Numerical Approximation.
Pergamon, 1966.

27. Hayes. J.G. Numerical Approximations to Functions and Data.
Athlone Press, 1970.

28. Van Warrington. Protective Relays Their Theory and Practice.
Vols. 1 and 2. Chapman and Hall, 1962.

29. Reyrolle Protective Systems - A Quick Reference List.
A. Reyrolle and Co. Ltd.

30. Power System Protection. Vols. 1, 2 and 3.
Edited by the Electricity Council. Macdonald, 1969.

31. Stigant. S.A., and Franklin. The J and P Transformer Book. 10th Ed.
Newnes - Butterworth, 1973.

APPENDIX 1

(CHAPTER 5)

USERS MANUAL

5.1 Introduction

This program can be used to perform the following analyses:

a) Balanced Load Flow.

b) Asymmetric fault analysis.

c) Protection performance under asymmetric fault conditions.

The program is written in Fortran IV using the extended Fortran facilities offered by I.C.L., this version of Fortran being compatible with I.B.M. Fortran IV. However, two machine oriented routines have been used - DEFBUF and COMP, see 4.22. If these routines are replaced by their equivalents, for different makes of computer, the program will function on any machine that has a compatible Fortran compiler and sufficient storage facilities.

5.2. Program Requirements

a) 26 700 words of immediate access store.

b) 4 independent overlay areas, 10 000 words each.

c) 10 common areas, the two largest of which require 11 000 and 7 000 words respectively. The remaining areas are less than 1 000 words each.

d) 12 disc files, totalling 534 records - 118 560 words.

5.3 Network Size Limitations

a) 3 to 40 nodes

b) 2 to 80 branches

c) 0 to 20 induction motors and/or synchronous machines

d) 0 to 20 relay positions - equal to $20 \times 3 = 60$ relays, I.D.M.T. and/or impedance relays.

e) 0 to 80 two winding transformers - off nominal tap positions possible.

f) 0 to 25 three winding transformers

It should be noted that the combined total of c, e and f is determined by $(c + e + (f \times 3 + 1)) = 80$

5.4 Data Control

There are 7 possible job control cards, which are used in various combinations to control the execution of each job. These cards are:

a) /*JOB NAME

b) /*JOB NUMBER

c) /*ANALYSIS

d) /*DATA

e) /*PRINT DATA

f) /*END

g) ////////// (10 slashes)

Note: the blanks in the above controls are mandatory.

The job control cards are used as follows:

a) to supply an identifying name to a job, columns 17 to 72 are available. If a job number is not specified, this card must be included.

b) to supply, or indicate, the identifying job number. The job number should be punched in columns 41 to 43, in Format I3.

c) to specify the type of analysis, the type of fault, the fault bus and the maximum network time, if required. The analysis is indicated by punching any of

LOAD FLOW
FAULT
PROTECTION

starting in column 17. The fault type is specified by punching any of

L/E	(for line to earth fault)
L/L	(for line to line fault)
L/L/E	(for line to line to earth fault)

starting in column 49. The number of the bus bar at which the fault is to occur is punched in columns 61 to 63, Format I3. The value of the maximum network time in seconds is punched in columns 66 to 72, Format F8.4.

d) indicates that the network data is to be supplied, on cards, at execution time.

f) signifies the end of the job control sequence.

g) signifies the end of the relay data, i.e. the last data card for a protection study.

5.5. Network Data Cards

The following is an ordered list of all possible data cards, and the card sequence for any given job is obtained by omitting those cards which are not required, see 5.6. for examples.

a) Format E20.6 , 3I3

The variable names and their function are:

TOL this is the tolerence to which the nodal voltages are found by the iteration procedure. A default value of 1×10^{-6} is applied.

LEVEL can be used to activate the programs internal trace mechanism, as explained in 6.44.

SCIND is used to specify the number of bus bars at which the symmetrical three phase short circuit is to be applied.

IPRINT is held in common area T4 and can be used to increase the amount of output that is printed. If IPRINT is set equal to 2 the values of the nodal phase voltages and the line currents will be printed at the end of each sub-analysis. Each complete analysis which involves the curve fitting routines consists of a minimum of 3 sub-analyses and one comprehensive analysis, and this procedure is repeated each time the network is modified because of relay operation.

b) Format 20I3

The numbers of the nodes for the symmetrical three phase short circuit, the total of which is specified by SCIND

c) Format 6I3

BB the number of system branches.

N the number of system nodes.

TF the number of two winding transformers with off-nominal turns ratios.

ICONT specifies the number of nodes at which the voltage is specified completely, excluding the swing bus. Typically these will be the machine emfs, the values of which were computed during a previous analysis.

ISWGN is set equal to 1 when bus 1 is not required to act as the swing bus, but is replaced as reference bus by one specified under ICONT.

IEMF is used for a load flow analysis when it is desired to compute the values of the emfs acting behind the machine reactances, see ICONT above. IEMF is set equal to the total number of nodes for which this is required.

d) N cards of nodal data.

Format 3I4, 5F10.4

M the node number.

ND7 the nodal designation code, 1 for a PQ bus, -1 for a PV bus and 0 for a reference bus.

MPS takes a value of zero or 1, see 5.

p the specified value of injected nodal power.

Q the specified value of injected nodal reactive power.

MODV the specified value of the modulus of the nodal voltage.

YL the modulus of the value of shunt admittance connected to node M.

ANGLE the argument associated with YL, degrees.

e) If ICONT is specified, this position is occupied by ICONT cards, each of which specify a node number and the value of the voltage at that node, in cartesian form, $V_p + jV_q$.

Format I3, 2F10.6

f) BB cards of branch data, see (c) above.

Format 2I4 2F10.4

K node number

M node number

R the value of the branch resistance.

X the value of the branch reactance.

g) If TF is specified this position is occupied by TF cards, each indicating a transformer location and its off nominal turns ratio.

Format F10.4 2I3

T1 the off nominal turns ratio.

K node number.

M node number.

h) If IEMF = k, this position is occupied by k cards as follows:

Format 2I3, 4F10.4

k the number of the node to which the machine will be connected.

IC the number of the machines internal node, see Fig. 5.1.

R the value of the machines internal resistance.

X the value of the machines internal reactance, usually the synchronous reactance.

P the value of power supplied by this machine.

Q the value of the reactive power supplied by the machine. P and Q are only required if there is more than one machine connected to bus bar K.

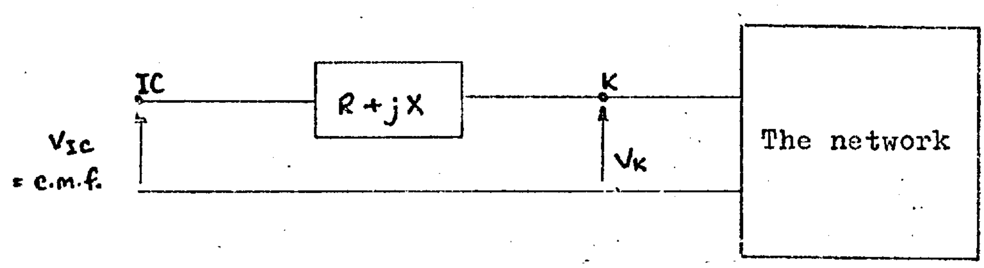


Fig. 5.1.

i) The fault impedance (asymmetric faults only),

Z_{fault} = R + jX.

Format 2F10.4

R the value of resistance.

X the value of reactance.

j) The number of negative sequence network modifications.

Format I3

ICCNS the number of modifications.

k) If ICCNS is specified this position is occupied by ICCNS cards each containing one new value of branch impedance.

Format	2I3 2F10.4
K	node number.
M	node number.
R	the value of resistance.
X	the value of reactance.

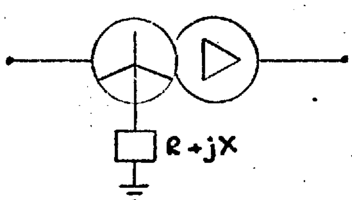
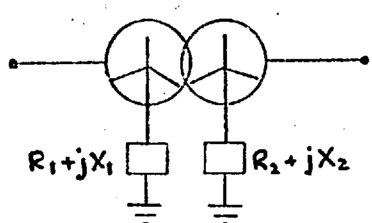
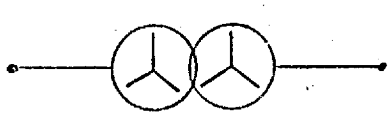
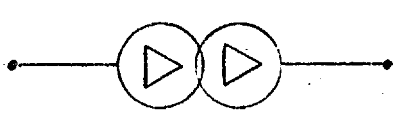
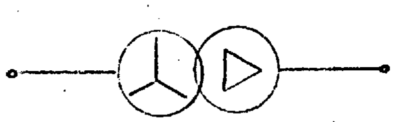
l) The number of zero sequence network modifications.

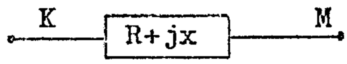
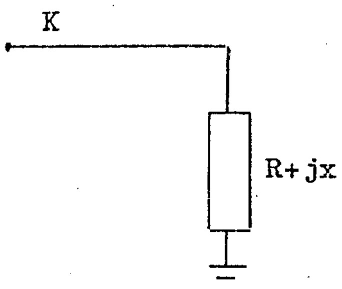
Format	I3
ICCZS	the number of modifications.

m) If ICCZS is specified this position is occupied by ICCZS cards, each as follows:

Format	3I3 4F8.4 3I3
K	node number
M	node number.
TFCON	the zero sequence network modification code, as specified below.
ZR	the value of resistance.
ZX	the value of reactance.
XZS	the value of the secondary winding equivalent reactance.
XZT	the value of the tertiary winding equivalent reactance.
MN	the middle node number.
NT	the tertiary node number.

As can be seen from (m) above, each of the zero sequence modification cards can carry any of three different types of information. Type one relates to two winding transformers, type two relates to system branches and/or system nodes, both one and two using the same set of variables; K, M, TFCON, ZR and ZX. Type three relates to the three winding transformers and uses all of the variables listed under (m) above. The modification codes and the applicable circuit arrangements are:

Circuit Connections	Code (TFCON)	ZR	ZX
	1	3.R	3.X
	2	$3(R_1+R2)$	$3(X_1+X_2)$
	1	0	0
	1	0	0
	1	0	0

Circuit Connections	TFCON	ZR	ZX
	2	R	X
	3	R	X

For three winding transformers the equivalent circuits are as discussed in section 2.5.2.2. The variables corresponding to the elements of the equivalent circuit as shown by Fig.2.4 are:

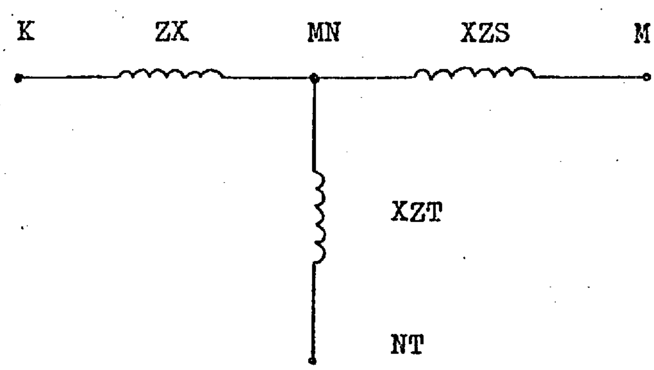


Fig. 5.3.

The modification codes and the corresponding circuit arrangements are shown below in Fig. 5.4.

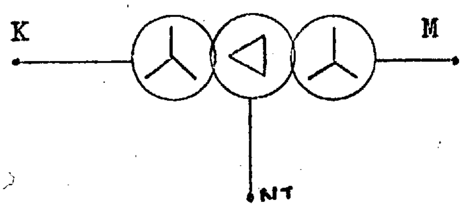
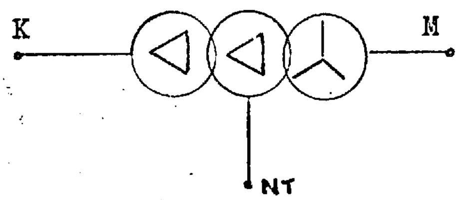
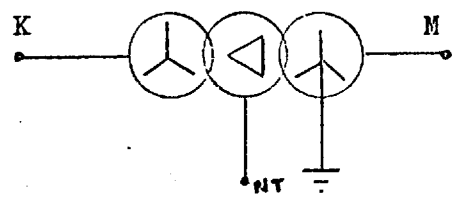
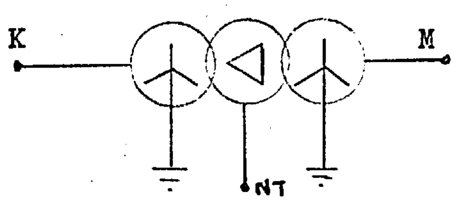
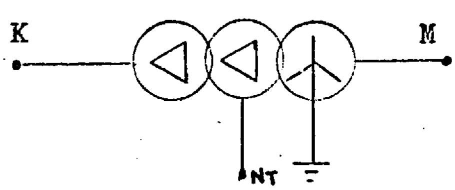
Connection Diagram	TFCON
	1
	1
	2
	3
	4

Fig. 5.4

n) When machine equivalent circuits, as discussed in section 2.5.3, are to be included in the network this area of data contains the machine parameters, one card for each machine.

Format	2I4 6F10.4
K	node number
M	node number
X1	sub transient reactance
X2	transient reactance
X3	synchronous reactance
TX1	the three time constants associated with the reactance values.
TX2	
TX3	

Note. The terminating card of this section is a BLANK card.

p) Relay data. Data for any given relay occupies several cards and therefore, requires careful assembly. To minimise errors, and to make error location easy, each relay data card has an identification code. The code includes the number of the relay, the phase in which the relay is located and the type of data supplied. In order to avoid a repetition of identical data, it is possible to access data that is already stored by inserting the letter F (file) in column 7, immediately before the phase indication. The number of the relay from which the data is copied is punched in columns 9 to 11, Format I3, and this relay will be a phase A relay. Obviously, any relay data which is to be used as a reference must be supplied before it can be accessed. Apart from this restriction relay data can be supplied in any order.

p.1 The number of relays

Format I3

p.2 The first 8 columns of each relay data card ~~is~~ are allocated as an identification area, containing the following information:

Columns	Data	Format
1 to 2	relay position	I2
3 to 6	data code	A4
7 to 8	phase and file	A2

There are four data codes:

▽IRL for cards supplying the node numbers which locate the branch where the relay is situated, and also the directional facility indication. IRL also supplies the intertrip information. That is, the numbers of the relays which will operate when the given relay operates. Each relay is limited to three intertrip indications.

▽▽CT for cards supplying the factors which affect the relay operating times, as calculated from the stored characteristic.

▽CF1 and ▽CF2 for cards which contain the relay characteristic information.

Where ▽ indicates a blank column.

The data supplied under each code is as follows:

Code	▽IRL	
Format	8X, 3I3	
Data	K	node number
	M	node number
		the third number is the directional indication,
	1	the relay 'looks' from K to M
	-1	the relay 'looks' from M to K
	0	non-directional
	2	impedance relay, direction K to M
	3	impedance relay, direction M to K.
Format	17X, 3I3	
Data	the numbers of up to 3 relays which will operate under intertrip when the specified relay, Columns 1 and 2, operates.	

Code	▽▽CT	
Format	8X, 4F10.4	
Data	CT1	the time scale factor, normally 1.0
	CT2	the plug setting multiplier
	CT3	the relay CT ratio.
	CT4	the time multiplier

Code	∇ CF1	(I.D.M.T. relays)
Format	8X, 4E15.6	
Data		the first four coefficients of the rational function representing the relay characteristic.
	∇ CF2	the second four coefficients.

For impedance relays these two data codes are used to supply the impedance values and the time for each zone.

Code	∇ CF1	(when used for impedance relays)
Format	8X, 4E15.6	
Data	C1	$\lvert Z \rvert$ for the first zone.
	C2	the time limit for the first zone.
	C3	$\lvert Z \rvert$ for the second zone.
	C4	the time limit for the second zone.

Code	∇ CF2	
Format	8X, 4E15.6	
Data	C1	$\lvert Z \rvert$ for the third zone.
	C2	the time limit for the third zone.

5.6 Specimen Problem

The following example has been chosen to clearly illustrate the network data requirements and the line printer output obtained from a protection performance study.

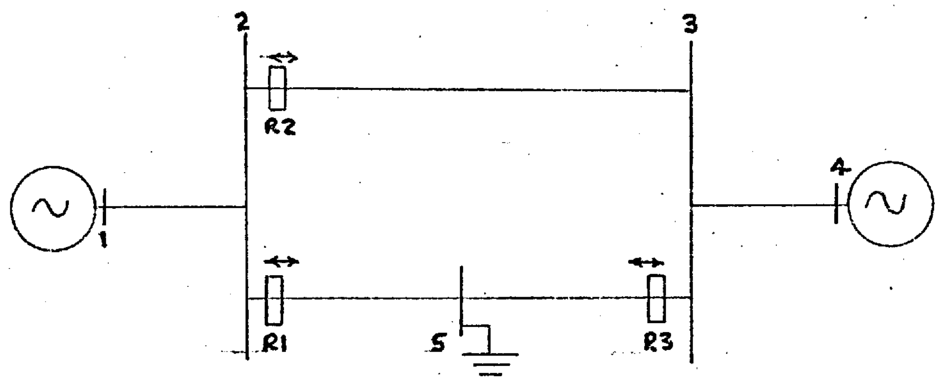


Fig. 5.5.

The relays R1, R2 and R3 have the same time/current characteristic - English Electric type CDG13 Very Inverse Overcurrent/Earth Fault relay. The system is initially unloaded when a single line to earth fault occurs at bus bar 5. The system data is specified by the following annotated list of data cards.

Input data	Description

`/*JOB NAME      DAVEY TEST SYSTEM 2.   IDMT RELAYS.`
`/*JOB NUMBER                           6`
`/*ANALYSIS      PROTECTION                L/E      5    0.25`
`/*DATA`
`/*END`
	Data control

`1.0E-6`	Telerance
`5 5`	The number of branches and number of nodes

`1              1.0`
`2  1  0`
`3  1  0`
`4              1.0`
`5  1  0`
	Nodal data

`1  2  3   0.02`
`2  3      0.1`
`3  4      0.02`
`2  5      0.05`
`3  5      0.05`
	Branch data

`2`	The number of network modifications for creation of N.S.N.
`1  2  0.0  0.14` `3  4  0.0  0.14`	Network modifications
`5`	The number of network modifications for creation of Z.S.N.

`3  5  2  0.0  0.1`
`2  5  2  0.0  0.1`
`2  3  2  0.0  0.2`
`3  4  2  0.0  0.1`
`1  2  2  0.0  0.1`
	Network modifications

`1  2  0.14  0.23  1.2  0.0  0.09  2.0` `4  3  0.14  0.23  1.2  0.0  0.09  2.0`	Machine data: reactances and times
`0`	Terminator

```
3
```

The number of relays

```
1 IRL A  2  5  0
1 IRL B  2  5  0
1 IRL C  2  5  0
```

Relay number 1. Location and directional indication (this is a non-directional relay)

```
1  CT A   1.0      1.0      1.0      0.1
1  CT B   1.0      1.0      1.0      0.1
1  CT C   1.0      1.0      1.0      0.1
```

Scaling factors, including time multiplier and P.S.M.

```
1 CF1 A  0.02398546    -0.05885443    0.04546986    -0.003890020
1 CF1 B  0.02398546    -0.05885443    0.04546986    -0.003890020
1 CF1 C  0.02398546    -0.05885443    0.04546986    -0.003890020
1 CF2 A  0.0001629406   -0.3213780E-05
1 CF2 B  0.0001629406   -0.3213780E-05
1 CF2 C  0.0001629406   -0.3213780E-05
```

Relay characteristic equation coefficients

```
2 IRL A  2  3  0
2 IRL B  2  3  0
2 IRL C  2  3  0
2  CT A   1.0      1.0      1.0      0.1
2  CT B   1.0      1.0      1.0      0.1
2  CT C   1.0      1.0      1.0      0.1
2 CF1 A  0.02398546    -0.05885443    0.04546986    -0.003890020
2 CF1 B  0.02398546    -0.05885443    0.04546986    -0.003890020
2 CF1 C  0.02398546    -0.05885443    0.04546986    -0.003890020
2 CF2 A  0.0001629406   -0.3213780E-05
2 CF2 B  0.0001629406   -0.3213780E-05
2 CF2 C  0.0001629406   -0.3213780E-05
```

Data for relay number 2

```
3 IRL A  3  5  0
3 IRL B  3  5  0
3 IRL C  3  5  0
3  CT A   1.0      1.0      1.0      0.05
3  CT B   1.0      1.0      1.0      0.05
3  CT C   1.0      1.0      1.0      0.05
3 CF1 A  0.02398546    -0.05885443    0.04546986    -0.003890020
3 CF1 B  0.02398546    -0.05885443    0.04546986    -0.003890020
3 CF1 C  0.02398546    -0.05885443    0.04546986    -0.003890020
3 CF2 A  0.0001629406   -0.3213780E-05
3 CF2 B  0.0001629406   -0.3213780E-05
3 CF2 C  0.0001629406   -0.3213780E-05
```

Data for relay number 3

```
////////
```

Data terminator

5.6.1 Specimen Line Printer Output

The following pages contain the line printer output for the problem specified in 5.6. Annotations have been added in order to clarify some of the detail.

```
       14/05/75                          08/53/31
*JOB NAME          DAVEY TEST SYSTEM 2.   IDMT RELAYS.
*JOB NUMBER                               6
*ANALYSIS       PROTECTION                    L/E          5   0.25
*DATA
*END
   /*JOB NAME           DAVEY TEST SYSTEM 2.   IDMT RELAYS.
```

FOR THIS STUDY

THE VOLTAGE' TOLERANCE IS   0.100000E-07

THE TRACE LEVEL IS   0

A SHORT CIRCUIT STUDY WILL BE MADE AT   0 BUSBARS

P	Q	MODV**2	MOD AND ANGLE OF Y		VP	VQ	ND7	MPS	J	ND	NDZ
0.0000	0.0000	1.0000	0.0000	0.0000	1.0000	0.0000	0	0	1	-1	-1
0.0000	0.0000	0.0000	0.0000	0.0000	0.8990	0.0000	1	0	2	1	1
0.0000	0.0000	0.0000	0.0000	0.0000	0.8990	0.0000	1	0	3	1	1
0.0000	0.0000	1.0000	0.0000	0.0000	1.0000	0.0000	0	0	4	-1	-1
0.0000	0.0000	0.0000	0.0000	0.0000	0.8990	0.0000	1	0	5	1	1

5	5	0	0	0	0
NB	N	TF	ICONT	ISWGN	IEMF

BRANCH DATA

BRANCH		RESISTANCE	REACTANCE
1	2	0.0000	0.0200
2	3	0.0000	0.1000
3	4	0.0000	0.0200
2	5	0.0000	0.0500
3	5	0.0000	0.0500

15 ITERATIONS

Balanced load flow results

NET GENERATIONS OR LOAD AT BUSBARS

	P.U.VOLTAGES		P.U. GENERATION		P.U. LOAD	
BUS	MOD	ANGLE (DEG)	MW	MVAR	MW	MVAR
1	1.0000	0.00			0.0000	-0.0000
2	1.0000	0.00			0.0000	0.0000
3	1.0000	0.00			0.0000	0.0000
4	1.0000	0.00			0.0000	-0.0000
5	1.0000	0.00			0.0000	0.0000

LINE FLOWS. (ALL VALUES IN P.U.)

LINE	SEND		RECEIVE		CURRENT	
	MW	MVAR	MW	MVAR	MOD	ANGLE
1 - 2	0.0000	0.0000	0.0000	0.0000	0.0000	0.00
2 - 3	0.0000	-0.0000	0.0000	-0.0000	0.0000	0.00
2 - 5	0.0000	-0.0000	0.0000	-0.0000	0.0000	0.00
3 - 4	0.0000	-0.0000	0.0000	-0.0000	0.0000	0.00
3 - 5	0.0000	0.0000	0.0000	0.0000	0.0000	0.00

```
1  5  0.0000  0.0000FAULT TYPE  FAULT BUS  FAULT IMPEDANCE
  2 THE NUMBER OF NEGATIVE SEQUENCE MODS

  1      2          0.000000E 00        -0.714286E 01     NEGATIVE SEQUENCE MODS

  3      4          0.000000E 00        -0.714286E 01     NEGATIVE SEQUENCE MODS
  5      1 FBUS      FAULT

 9       ITERATIONS FOR NEGATIVE NETWORK

   THE NEGATIVE SEQUENCE IMPEDANCE IS              0.0000  +  J(    0.0950 )
K M TFCON  ZR  ZX  XZS  XZT  MN  NT     3   5   2   0.0000   0.1000   0.0000   0.0000   0   0

K M TFCON  ZR  ZX  XZS  XZT  MN  NT     2   5   2   0.0000   0.1000   0.0000   0.0000   0   0

K M TFCON  ZR  ZX  XZS  XZT  MN  NT     2   3   2   0.0000   0.2000   0.0000   0.0000   0   0

K M TFCON  ZR  ZX  XZS  XZT  MN  NT     3   4   2   0.0000   0.1000   0.0000   0.0000   0   0

K M TFCON  ZR  ZX  XZS  XZT  MN  NT     1   2   2   0.0000   0.1000   0.0000   0.0000   0   0

 8       ITERATIONS FOR THE ZERO SEQUENCE NETWORK

         THE ZERO SEQUENCE IMPEDANCE IS            0.0000  +  J(    0.1000 )
                 0.000000E 00                 -0.512821E 01     5
```

Note – as seen from the fault position (beside the negative sequence impedance line)

Note – as seen from the fault position (beside the zero sequence impedance line)

14 ITERATIONS

POSITIVE SEQUENCE NETWORK MODIFICATIONS

REACTANCE AND TIME VALUES	1	2	0.1400	0.2300	1.2000	0.0000	0.0900	2.0000
REACTANCE AND TIME VALUES	4	3	0.1400	0.2300	1.2000	0.0000	0.0900	2.0000
REACTANCE AND TIME VALUES	0	0	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

26 ITERATIONS — First Analysis at t = 0.0 sec.

39 ITERATIONS — Second Analysis at t = 0.09 sec.

43 ITERATIONS

THIS STUDY HAS A MAXIMUM NETWORK TIME OF 0.25 SECONDS — Third Analysis at t = 0.25 sec (TMAX)

3 RELAYS

RELAY POSITION 3 PHASE 1 HAS A TRIP IN TIME 0.1360 SECONDS

LINE 3 5 OPENED
LINE 3 5 OPENED
LINE 3 5 OPENED

THE TIME VALUES ARE 0.136000 0.193000 0.250000

5 1 FBUS FAULT

12 ITERATIONS FOR NEGATIVE NETWORK

THE NEGATIVE SEQUENCE IMPEDANCE IS 0.0000 + J(0.1384) Note - modified N.S.N.

11 ITERATIONS FOR THE ZERO SEQUENCE NETWORK

THE ZERO SEQUENCE IMPEDANCE IS 0.0000 + J(0.1750) Note - modified Z.S.N.
0.0000000E 00 -0.319060E 01 5

27 ITERATIONS First Analysis at t= 0·136

31 ITERATIONS Second Analysis at t = 0·193

34 ITERATIONS Third Analysis at t= 0·25

RELAY POSITION 1 PHASE 1 HAS A TRIP IN TIME 0.2200 SECONDS

LINE 2 5 OPENED
LINE 2 5 OPENED
LINE 2 5 OPENED

THE TIME VALUES ARE 0.220000 0.235000 0.250000

5 1 FBUS FAULT

0.000000E 00 0.232831E-09

FAULT ISOLATED. RUN STOPPED

APPENDIX 2

(CHAPTER 6)

DETAILED PROGRAM DESCRIPTION AND FLOW CHARTS

Table of Contents - Appendix 2. | Page

6.1 Introduction

This appendix gives a detailed description of the program, so that when read in conjunction with chapter 5 the logic and operation of the program may be clearly understood. The appendix has five sections:

Section 1. The subroutines and functions used in the program and the relationships between these routines.

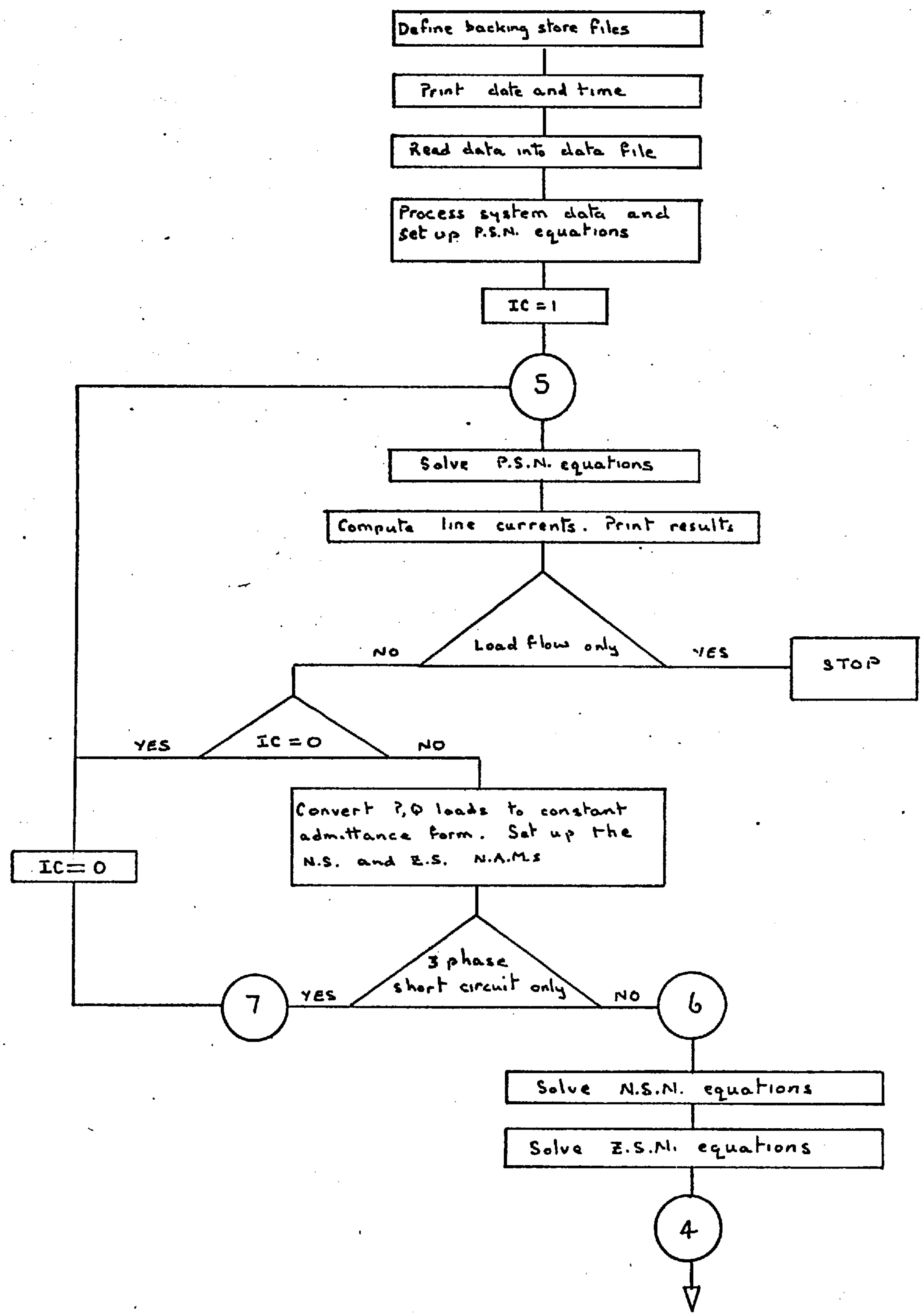
Section 2. Lists and descriptions of the various warning and error messages that can arise during the operation of the program. This section also details the operation of, and the results obtained from, the program trace mechanism.

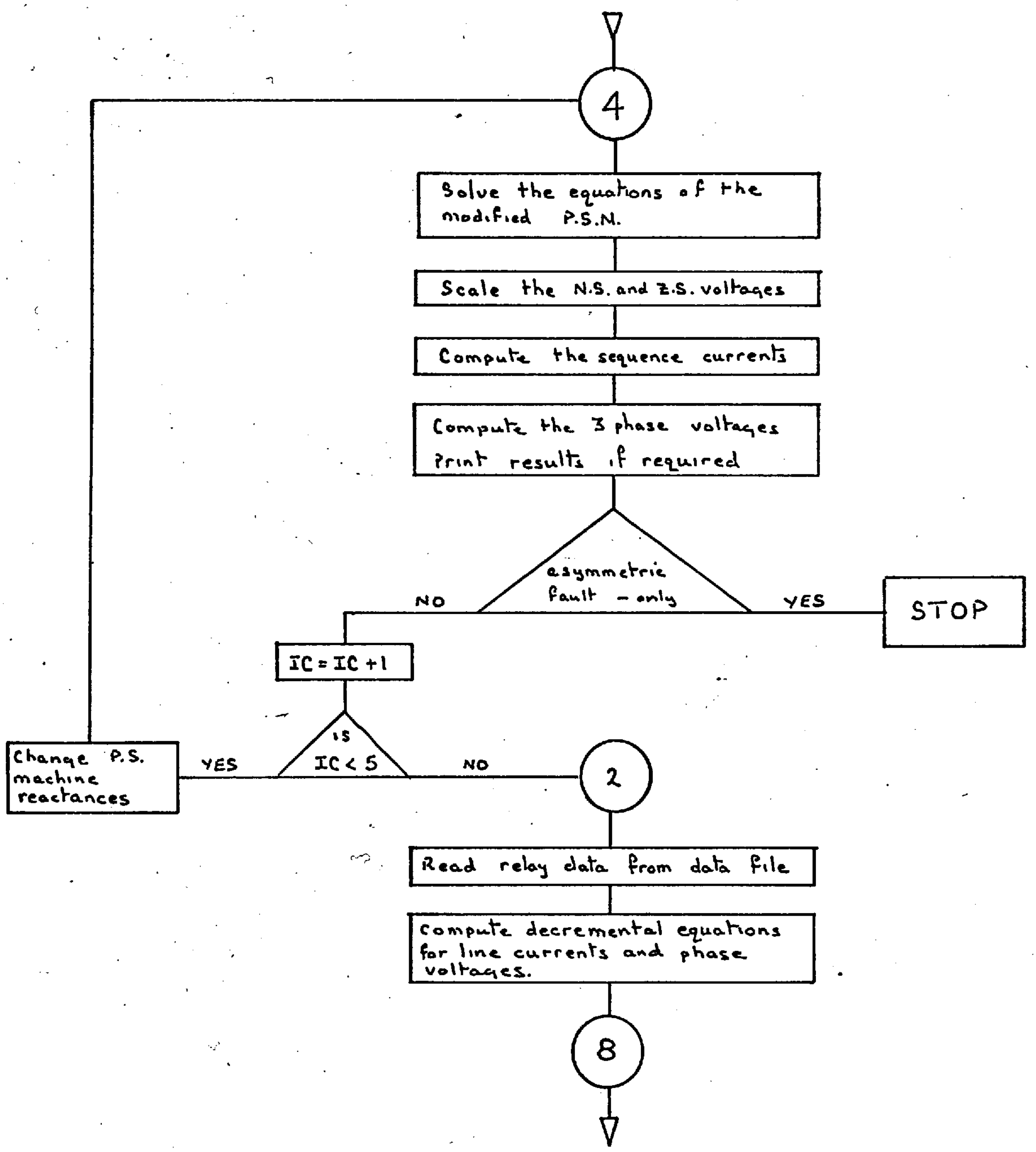
Section 3. The File Storage Scheme.

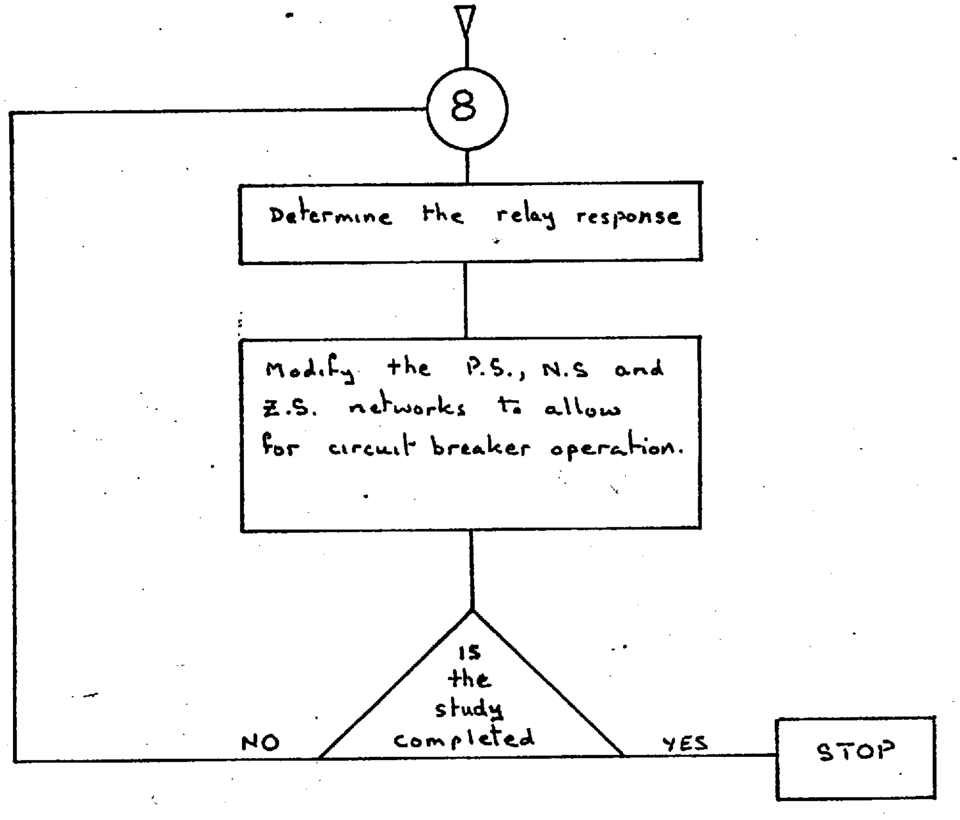
Section 4. The Overlay Scheme.

Section 5. A complete program listing.

6.2 Subroutine PSAA. Flow Chart




4
Solve the equations of the modified P.S.N.
Scale the N.S. and Z.S. voltages
Compute the sequence currents
Compute the 3 phase voltages
Print results if required
asymmetric fault - only
NO
YES
STOP
IC = IC + 1
is IC < 5
YES
NO
Change P.S. machine reactances
2
Read relay data from data file
Compute decremental equations for line currents and phase voltages.
8


8
Determine the relay response
Modify the P.S., N.S and Z.S. networks to allow for circuit breaker operation.
Is the study completed
NO
YES
STOP

6.3 Subroutine DATACNTRL

A job is run by supplying a complete set of system data, or by accessing system data which has been previously stored in a disc file. The type of job, that is the analysis required and the form of data input, are controlled by a series of 'job cards'. There are six possible job cards:

1. /*JOB NAME
2. /*DATA
3. /*JOB NUMBER
4. /*ANALYSIS
5. /*PRINT DATA
6. /*END

Subroutine DATACNTRL checks the job control cards to see if there is a compatible set of requirements. If system data is supplied, it is accessed by this routine and stored, if required, in an appropriate disc file. In order to identify these job control cards, a non-standard Fortran character manipulation routine is used. This routine, COMP+, is supplied by ICL++ as part of the software package with their 1900 series computers.

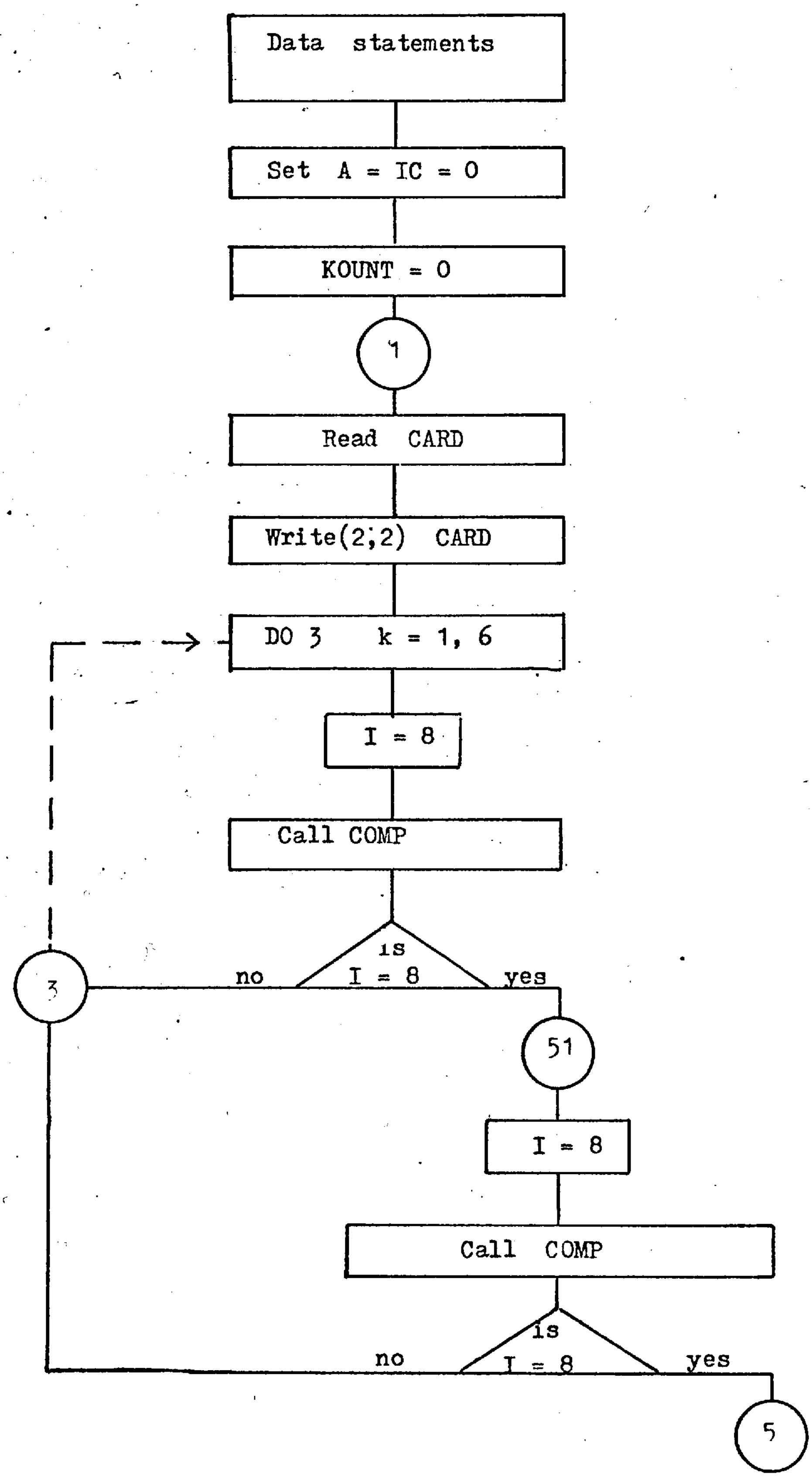
+ ICL technical publication 4314 2nd edition March 1972

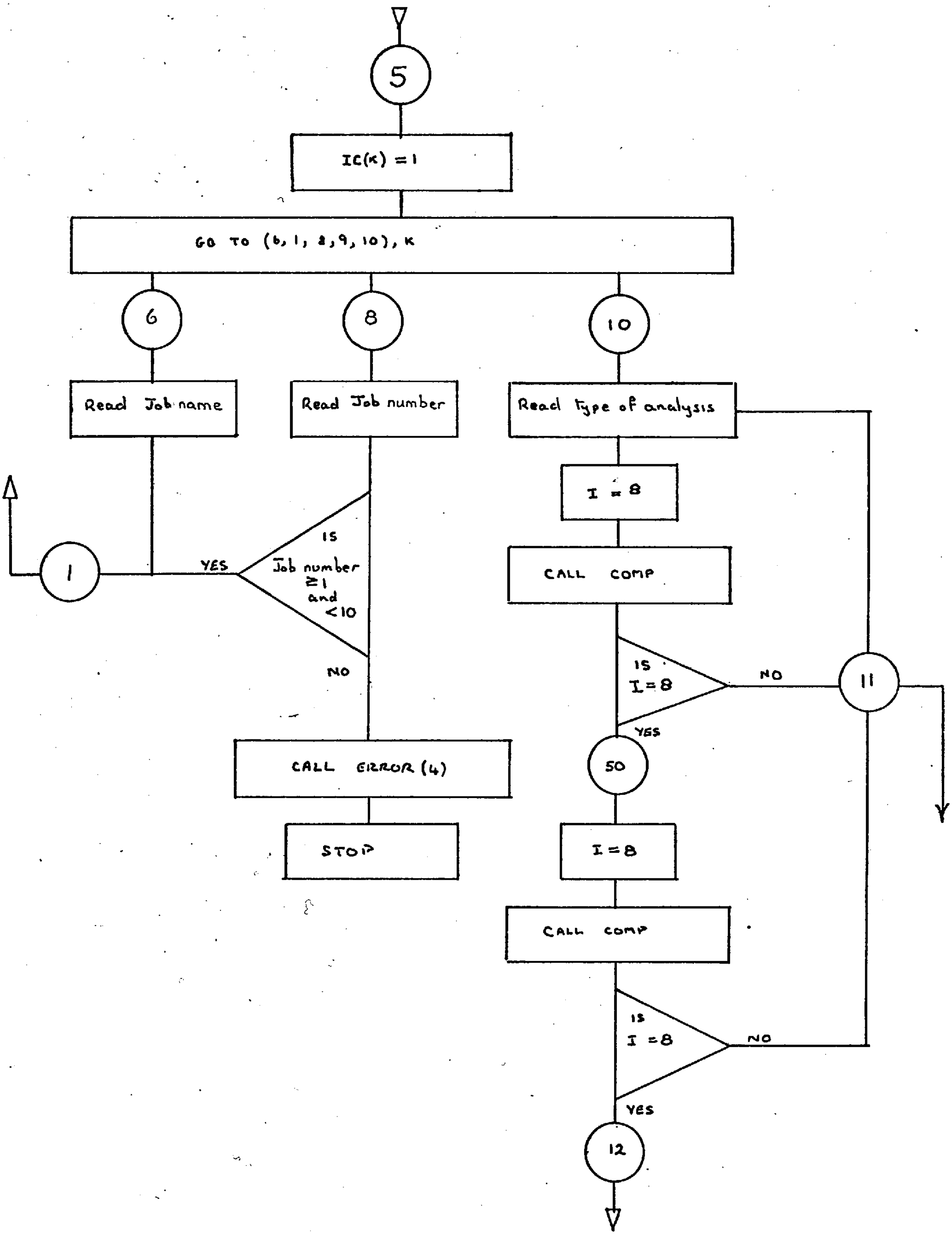
++ International Computers Limited.

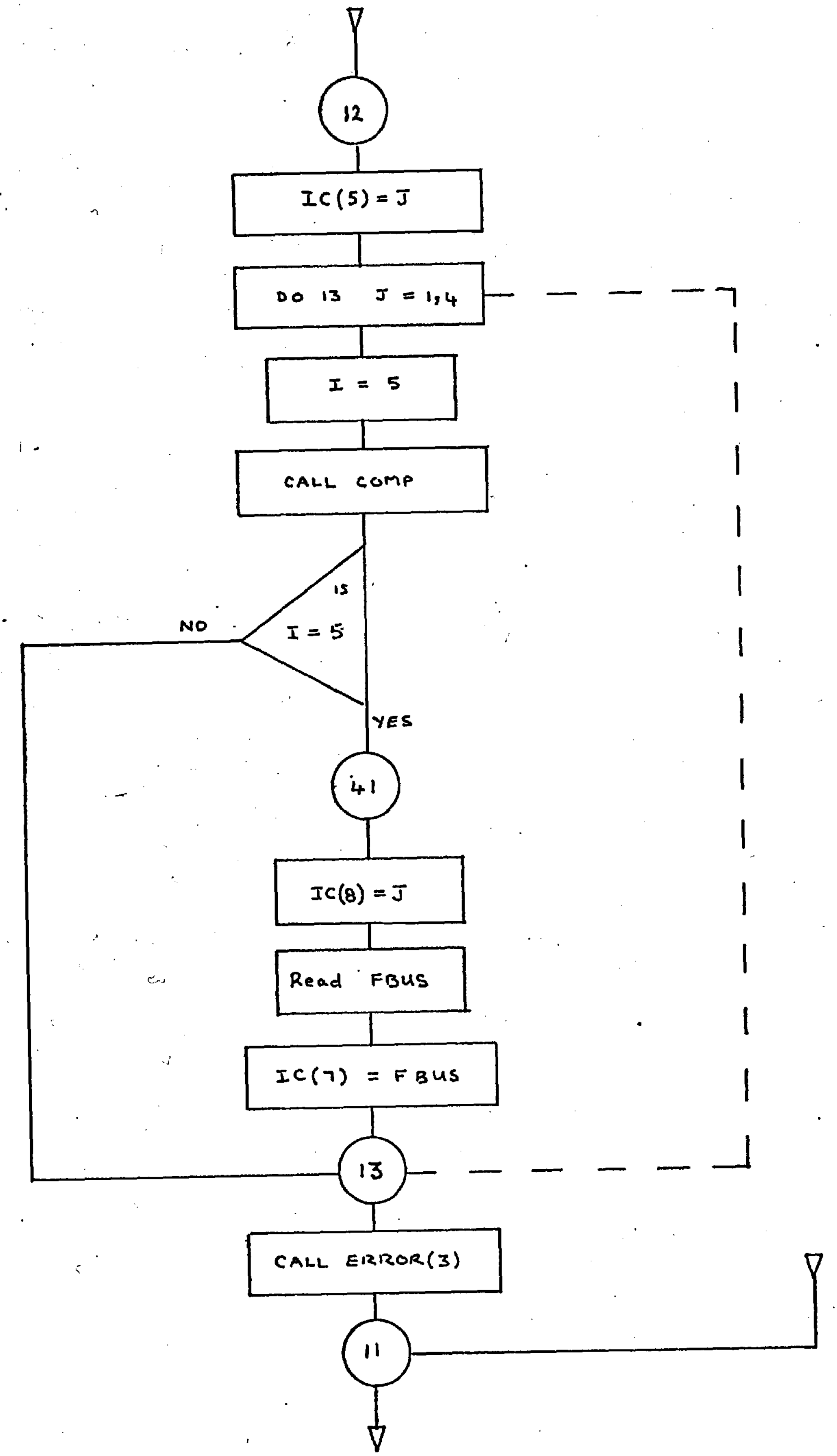
If the job control data is acceptable, this routine reads all data cards and stores the information - one card per record - in one of ten disc files. Disc file 1 is for temporary jobs, i.e. the next temporary job data overwrites the data of the preceding temporary job. Disc files 2 to 10 are used to store the job data on a more permanent basis, however, it is possible to subsequently modify the job control specification for these jobs and several different analyses may be performed by accessing the system data stored in any of the disc files 2 to 10. This facility is fully explained in the User Manual, see Chapter 5.

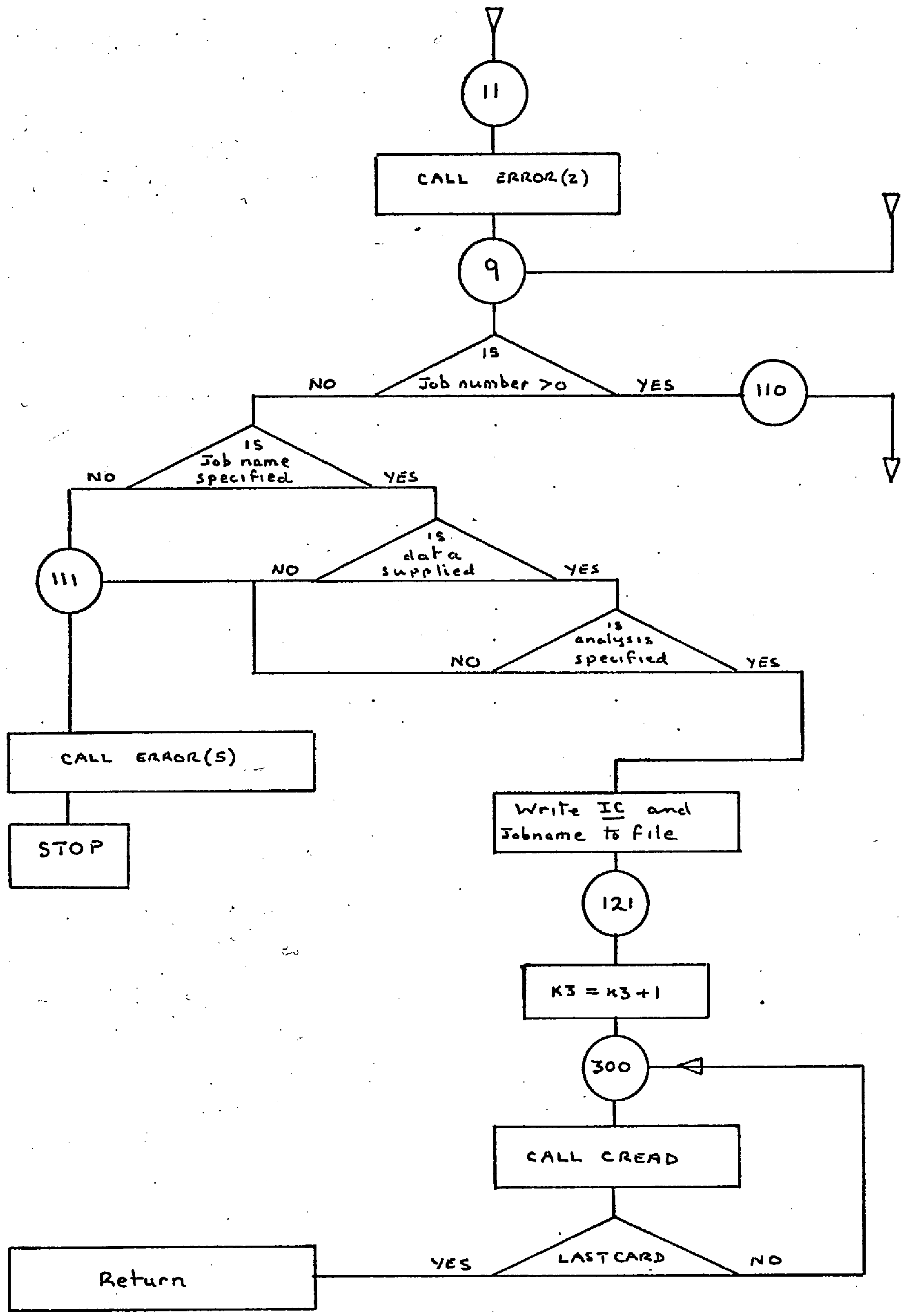
This is the only subroutine which reads cards, all other subroutines read the records (each record is effectively one card) which have been set up by DATACNTRL.

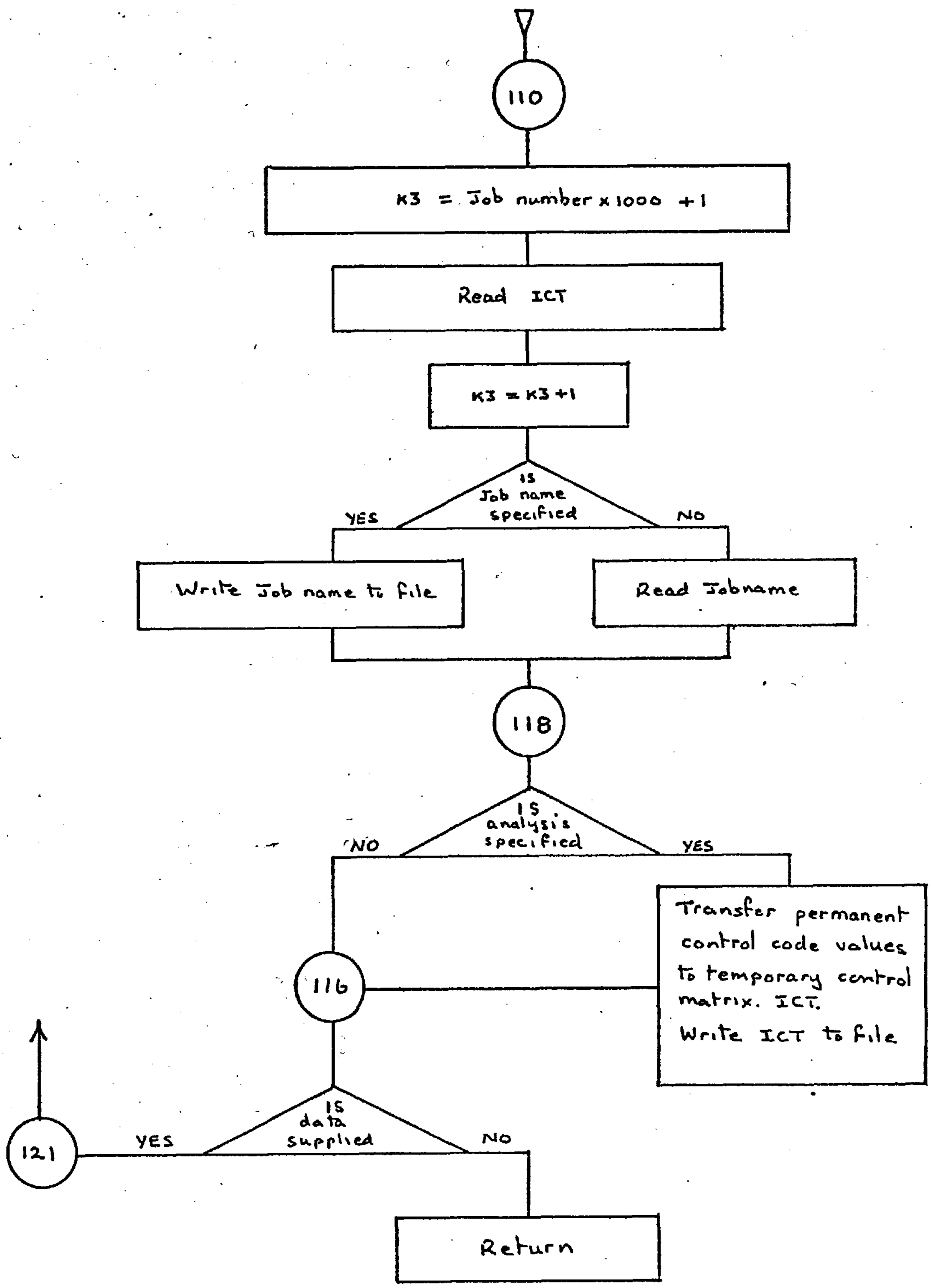
Subroutine Data Contrl




5
IC(K) = 1
GO TO (6, 1, 8, 9, 10), K
6
8
10
Read Job name
Read Job number
Read type of analysis
I = 8
CALL COMP
IS Job number ≥1 and <10
YES
NO
1
CALL ERROR (4)
STOP
IS I = 8
NO
YES
11
50
I = 8
CALL COMP
IS I = 8
NO
YES
12


12
IC(5) = J
DO 13 J = 1,4
I = 5
CALL COMP
IS
I = 5
NO
YES
41
IC(8) = J
Read FBUS
IC(7) = FBUS
13
CALL ERROR(3)
11


11
CALL ERROR(2)
9
IS
Job number >0
NO
YES
110
IS
Job name
specified
NO
YES
IS
data
supplied
NO
YES
111
IS
analysis
specified
NO
YES
CALL ERROR(5)
STOP
Write IC and
Jobname to file
121
K3 = K3+1
300
CALL CREAD
LAST CARD
YES
NO
Return


110
K3 = Job number x 1000 + 1
Read ICT
K3 = K3 + 1
IS Job name specified
YES
NO
Write Job name to file
Read Jobname
118
IS analysis specified
NO
YES
Transfer permanent control code values to temporary control matrix. ICT.
Write ICT to file
116
IS data supplied
YES
NO
121
Return

6.4 Subroutine PSA1

6.4.1 General Description

This subroutine reads the initial system control information and several types of system data, depending on the job control specification.

6.4.2 Initial System Control Data

The first three records contain the following information:

1. Job control matrix IC.
2. Job name, stored in matrix A2.
3. a. TOL — This is the tolerance to which the iterative solution will be computed.

   b. LEVEL — This variable can be used to obtain the state of some important variables at intermediate stages of the computation. (See 6.44)

   c. ISCIND — When the analysis to be performed is a Three Phase Short Circuit analysis this variable is set equal to the number of bus-bars at which the analysis is required.

   d. IPRINT — This variable is a print control cariable, and if used increases the amount of computed system information that is printed. (See 5.5)

6.4.2.1 If a three phase short circuit analysis is to be performed, this record will contain the numbers of the nodes at which the fault is to occur.

Otherwise, this record will contain the following data:

a. BB The number of system branches.

b. N The number of system nodes.

c. TF The number of transformers with off-nominal turns ratios.

d. ICONT This variable is used if machine emfs, computed in a previous load flow analysis, are to be used as system data.

e. ISWGN Bus number 1 is normally used as the swing bus, however if this is to be changed ISWGN is used.

f. IEMF This variable is used when it is required to compute the machine emfs, see d above.

6.4.3 System Data

The next N records, see b above, contains the nodal data, each record being made up as follows:

a. M The node number.

b. ND7(M) The nodal designation.

c. MPS(2,M) The relative nodal angle.

d. PS(M) The specified value of injected nodal power.

e. QS(M) The specified value of the injected nodal reactive power.

f. MODV(M) The specified value of voltage modulus.

g. YL(M) The modulus of the specified value of shunt admittance connected at the node.

h. ANGLE(M) The angle associated with g above.

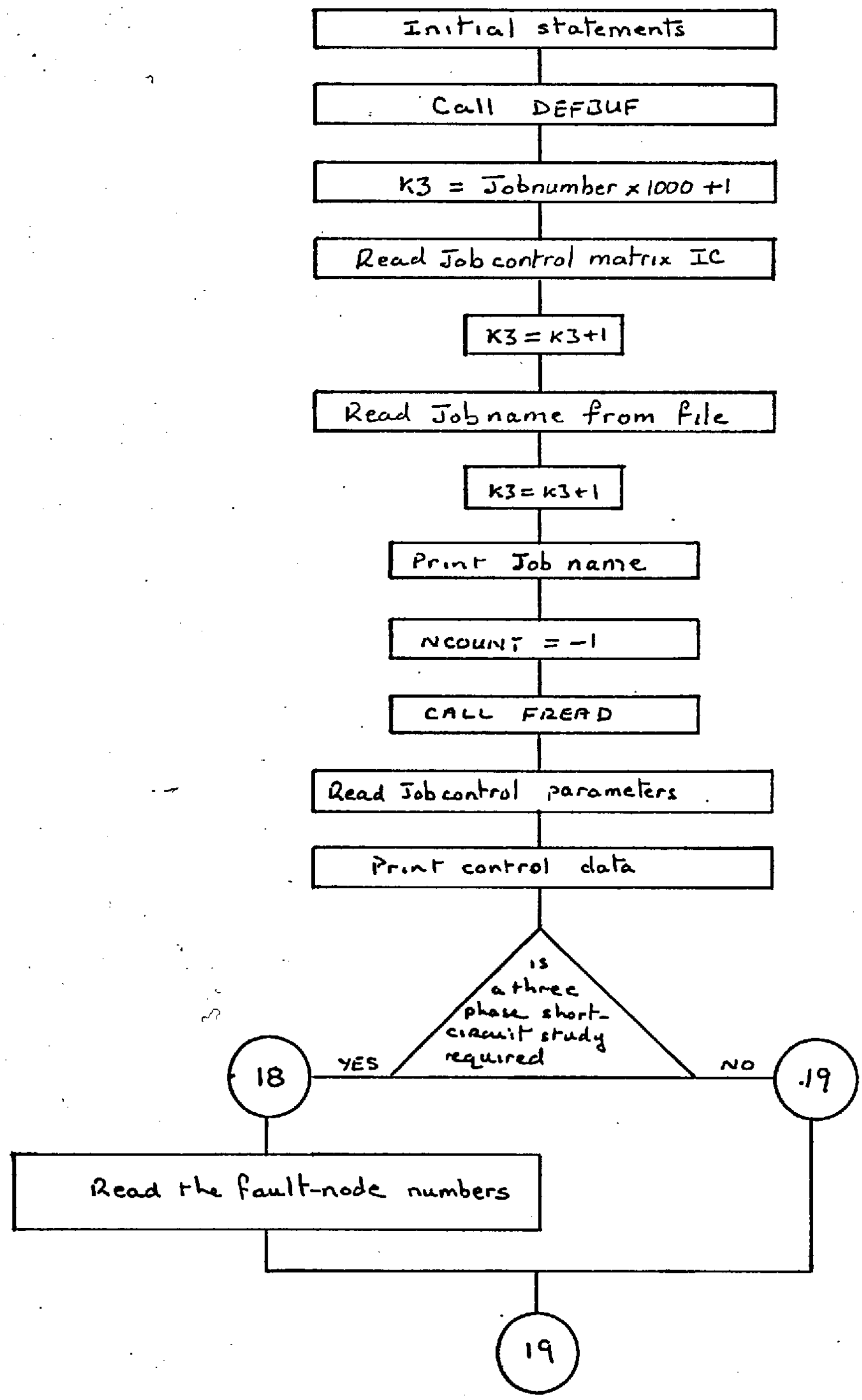
If ICONT is used, see 4.d above, the next ICONT records will each contain a node number and the real and quadrature components of the machine emfs.

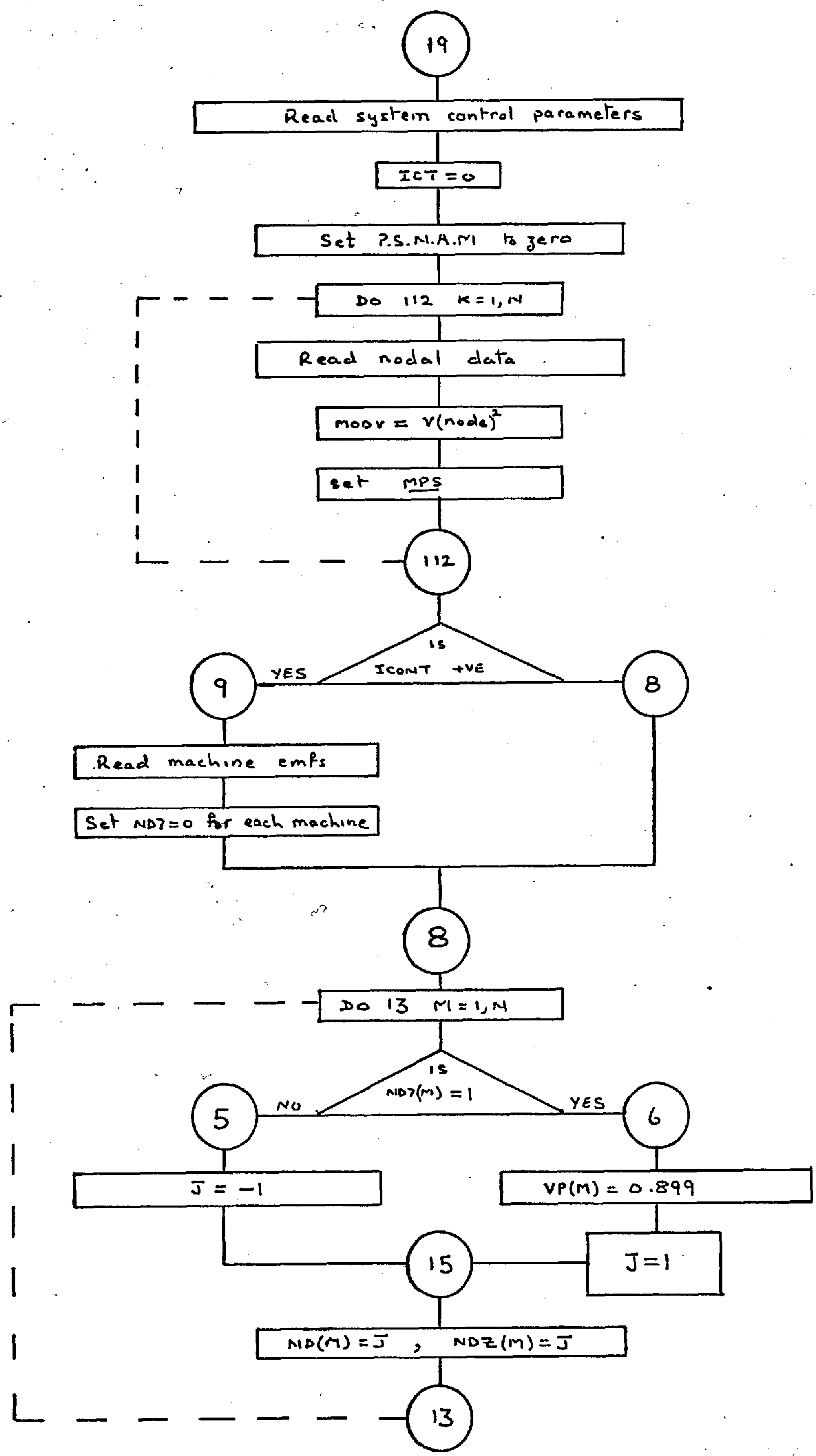
The above data is processed by subroutine PSA1 and allocated to the system data files, see 6.47.

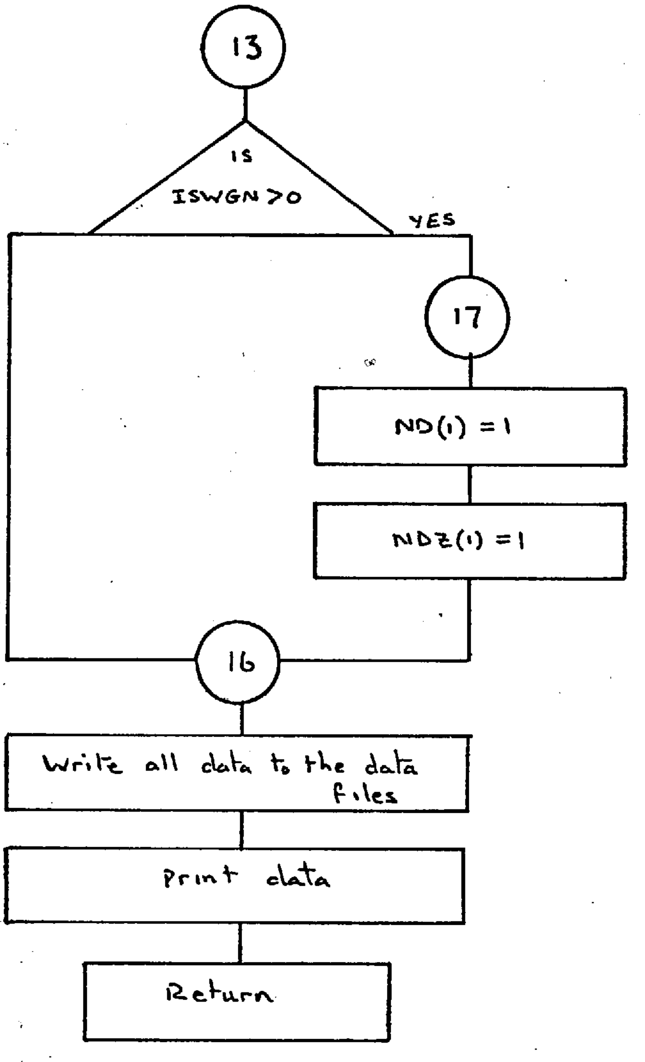
Note All values are in per unit, and loads are specified as negative generations. It is not possible to specify d, e and f at the same time, as they are not independent quantities. Only d and e, or d and f are permitted.

If the voltage modulus is not specified the initial value of the nodal voltage is set to 1.0 + j0, otherwise to $|V| + j0$. The elements of the nodal designation matrices take one of the following values -1 for a generator node (PV node), +1 for a load node (PQ node) and 0 for the reference node i.e. the swing bus. 0 is also used for nodes where the voltage is specified completely, as $V_p + jV_q$.

Subroutine PSA1 - Flow Chart




19
Read system control parameters
ICT = 0
Set P.S.M.A.M to zero
DO 112 K = 1, N
Read nodal data
MOOV = V(node)²
set MPS
112
IS ICONT +VE
YES
9
8
Read machine emfs
Set ND7 = 0 for each machine
8
DO 13 M = 1, N
IS ND7(M) = 1
NO
YES
5
6
J = −1
VP(M) = 0.899
15
J = 1
ND(M) = J , NDZ(M) = J
13

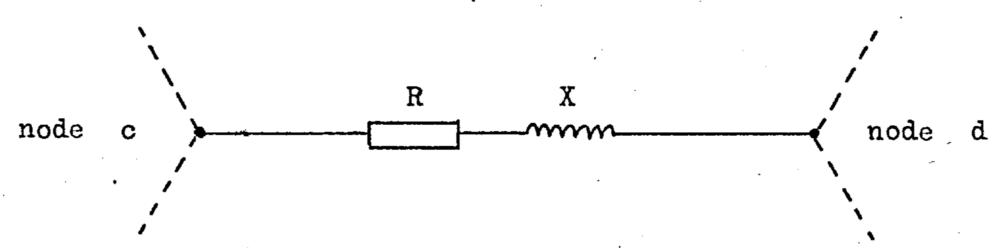

13
IS
ISWGN > 0
YES
17
ND(1) = 1
NDZ(1) = 1
16
Write all data to the data files
Print data
Return

6.5 Subroutine PSA2

6.5.1 General Description

This subroutine reads the branch data, and after converting the branch impedance value to an admittance value, places the real and quadrature components of this admittance in one of the columns of the branch matrix. The branch matrix is used subsequently as the basis from which the nodal admittance matrix is assembled.

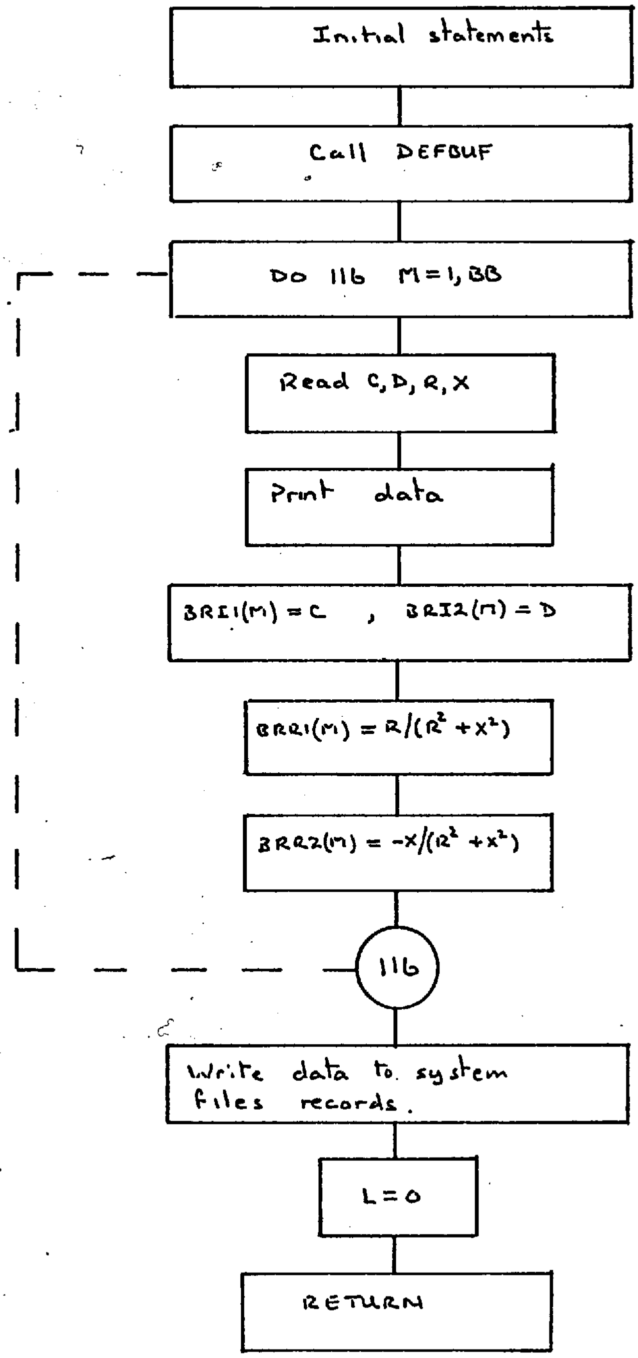
The branch matrix has 4 rows and BB columns, where BB is the number of branches in the system.



Branch K connected between nodes c and d.

When the Kth branch record is read by this subroutine the following information is stored in column k of the branch matrix: the numbers c and d that identify the nodes between which the branch is connected, are stored in rows 1, and 2 respectively. The branch impedance R + jX is inverted to form G + jB and the values G and B are stored in rows 3 and 4 respectively.

Subroutine PSA2 - Flow Chart



6.6 Subroutines PSA3 and PSA4

6.6.1 General Description

When a system contains more than 40 nodes, the nodal admittance matrix will be too large to hold in a 32 thousand word immediate access store along with the computing system executive software and the relevant parts of this program. This program was developed on such a machine, and to overcome this limitation only one row of the N.A.M. is held in the immediate access store at any one time.+ The N.A.M. is, therefore, assembled one row at a time.

The branch data, which is stored in the branch matrix, is usually compiled in random order and the data for each row of the N.A.M. has to be selected from the branch matrix as each row is assembled. The selection of the branch admittance values from the branch matrix, and the placing of these values in the N.A.M. is performed by subroutines PSA3 and PSA4, with the aid of an auxiliary matrix - SORT.

\+ When the number of nodes is less than 41, the iteration procedure works from a triangularised N.A.M. (see PSA7).

The matrix SORT is a row matrix of BB elements, which initially have the values 1 to BB, such that SORT(k) = k. The elements of this matrix are then manipulated, with reference to BRI1 (the first row of the branch matrix) so that the final arrangement of the elements of SORT indicates the positions of the ordered branch elements.

For example, consider the following branch matrix which represents a system which has 8 branches.

BRI1	5	1	4	2	1	2	2	5
BRI2	1	2	7	6	8	3	4	3
BRR1	G_{51}	G_{12}	G_{47}	etc.				
BRR2	B_{51}	B_{12}	B_{47}	etc.				

The initial values of the elements of SORT will be:

1	2	3	4	5	6	7	8

The final values of the elements of SORT will be:

2	5	4	6	7	3	1	8

This effectively changes the first two rows of the branch matrix to

1	1	2	2	2	4	5	5
2	8	6	3	4	7	1	3

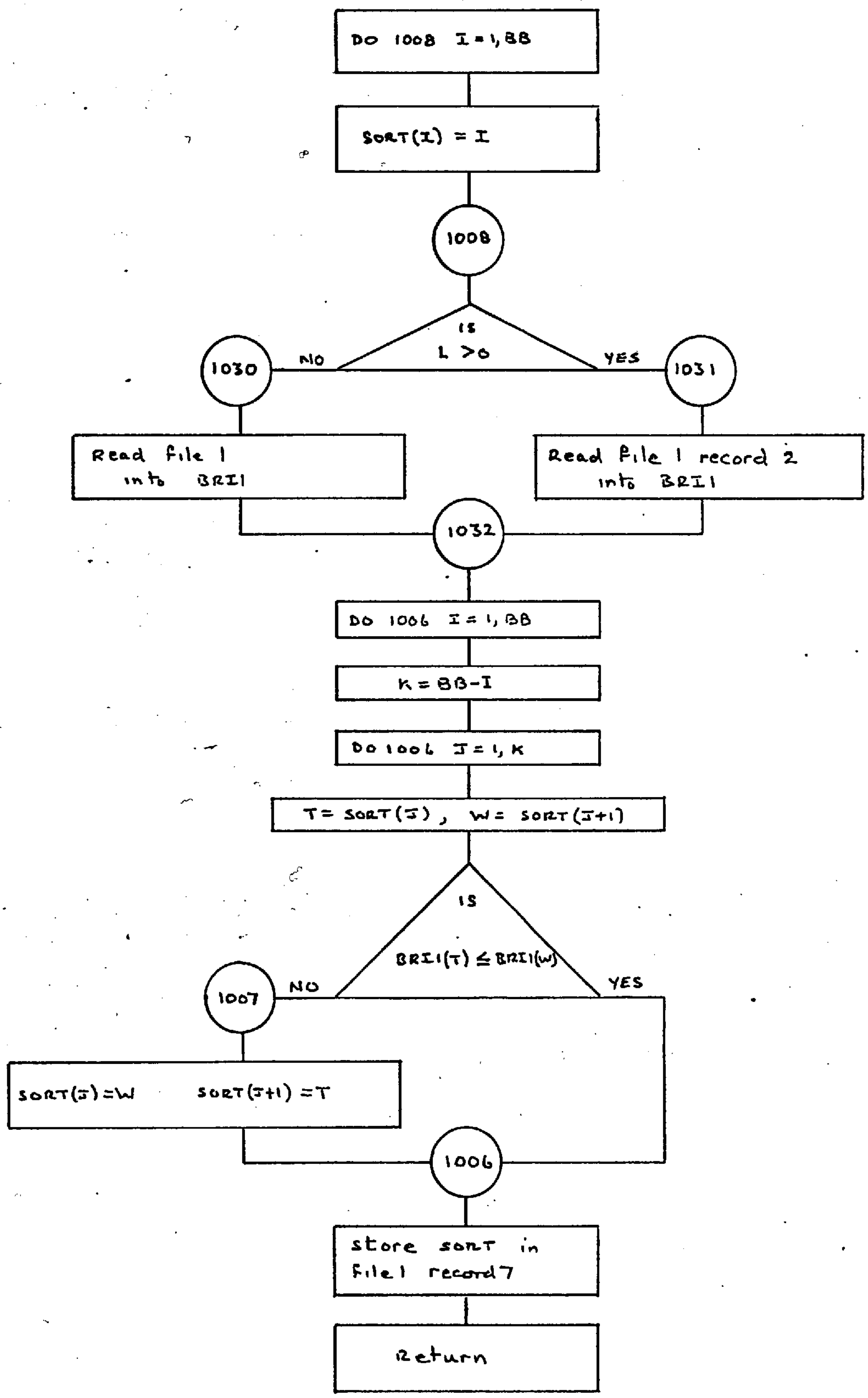
The N.A.M. is assembled as follows:

Row 1	insert elements	(1,2), (1,8)
Row 2	insert elements	(2,6), (2,3), (2,4)
Row 4	insert element	(4,7)
Row 5	insert elements	(5,1), (5,3)

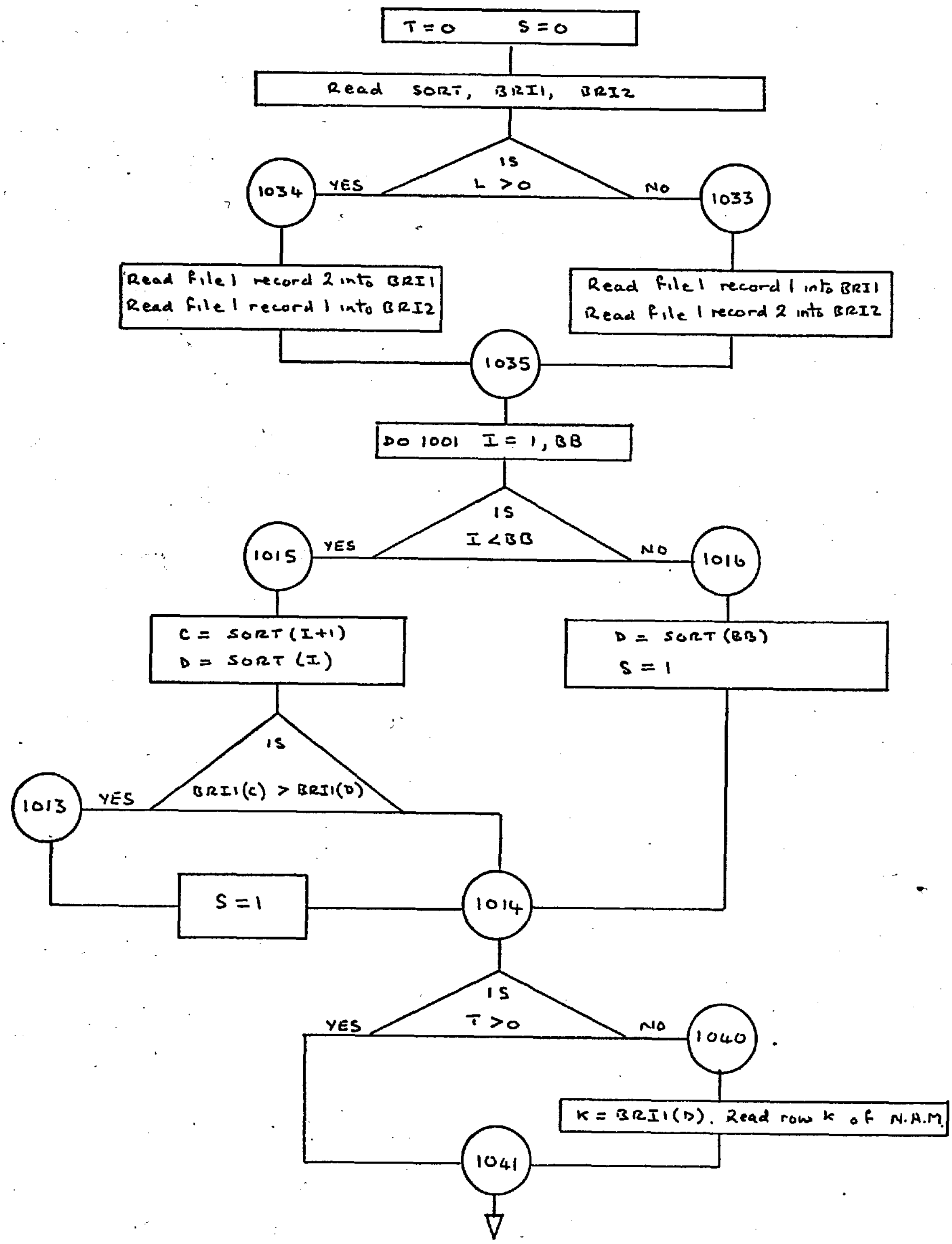
This procedure is accomplished by subroutine PSA4. Subroutine PSA3 is then called again, this time it operates on row 2 of the branch matrix - BRI2. PSA4 is called again, this time it inserts the remaining elements of the N.A.M. as listed below:

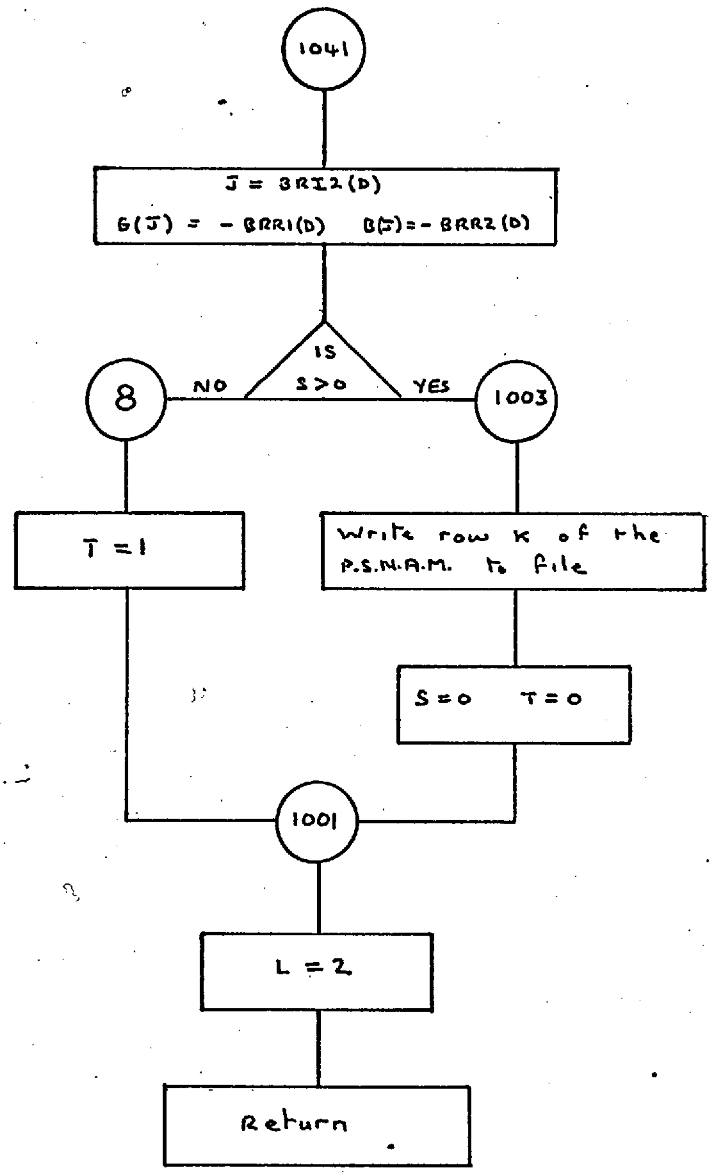
Row 1	insert element	(1,5)
Row 2	insert element	(2,1)
Row 3	insert elements	(3,2), (3,5)
Row 4	insert element	(4,2)
Row 6	insert element	(6,2)
Row 7	insert element	(7,4)
Row 8	insert element	(8,1)

Subroutine PSA3 - Flow Chart



Subroutine PSA4 - Flow Chart




1041
J = BRI2(D)
G(J) = - BRR1(D) B(J) = - BRR2(D)
IS S > 0
NO
YES
8
1003
T = 1
Write row K of the P.S.N.A.M. to file
S = 0 T = 0
1001
L = 2
Return

6.7 Subroutine PSA5

This subroutine transfers the shunt admittance values, computed from YL and ANGLE, to the appropriate diagonal elements of the N.A.M. The final values for the diagonal elements are then evaluated, firstly Y(1,1) and then Y(k,k), $k > 1$. The relevant equations are:

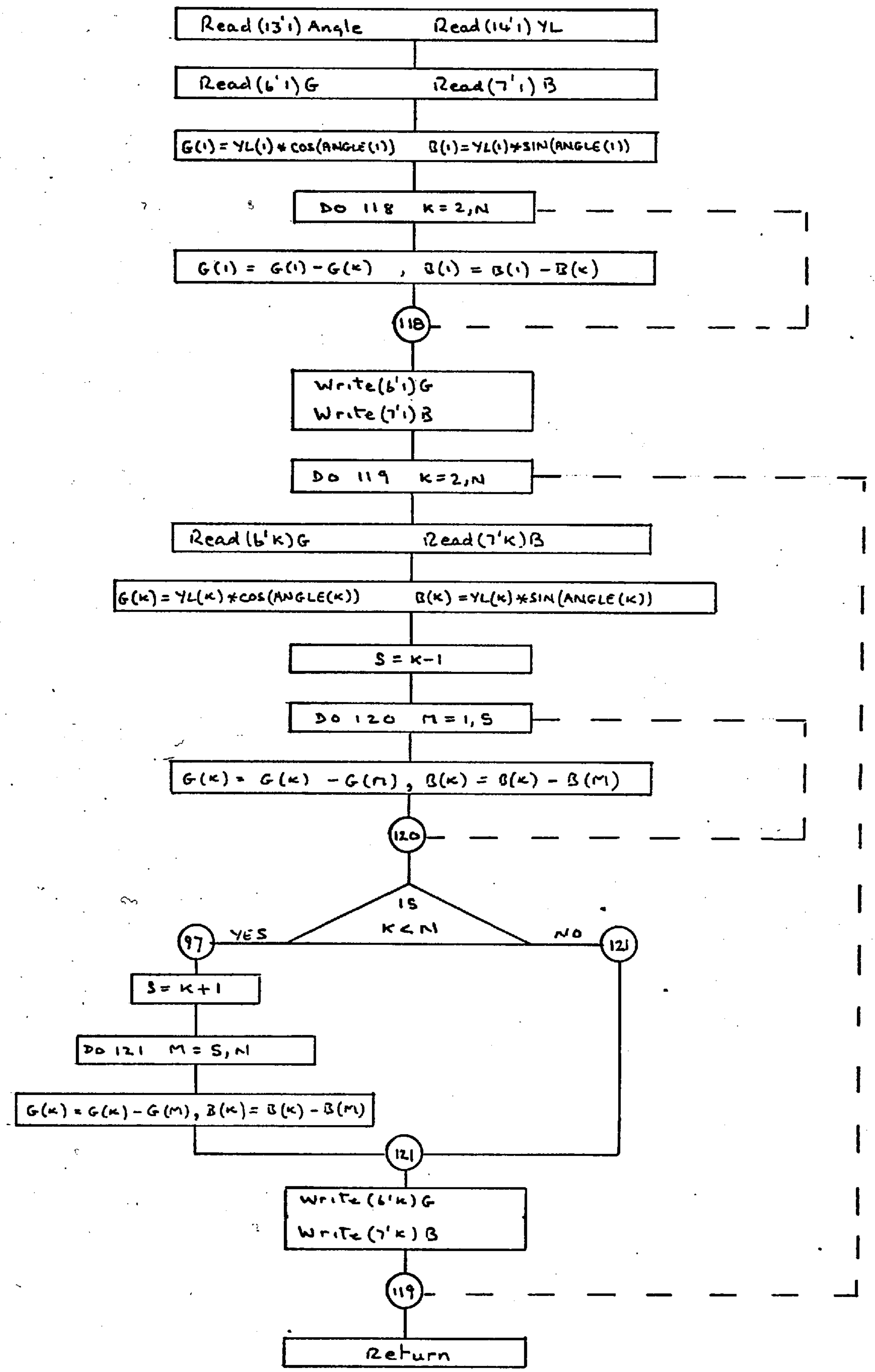
$$G(1,1) = G(1,1) - \sum_{m=2}^{n} G(1,m)$$

$$B(1,1) = B(1,1) - \sum_{m=2}^{n} B(1,m)$$

$$G(k,k) = G(k,k) - \sum_{\substack{m=1 \\ m \neq k}}^{n} G(k,m)$$

$$B(k,k) = B(k,k) - \sum_{\substack{m=1 \\ m \neq k}}^{n} B(k,m)$$

Subroutine PSA5 - Flow Chart

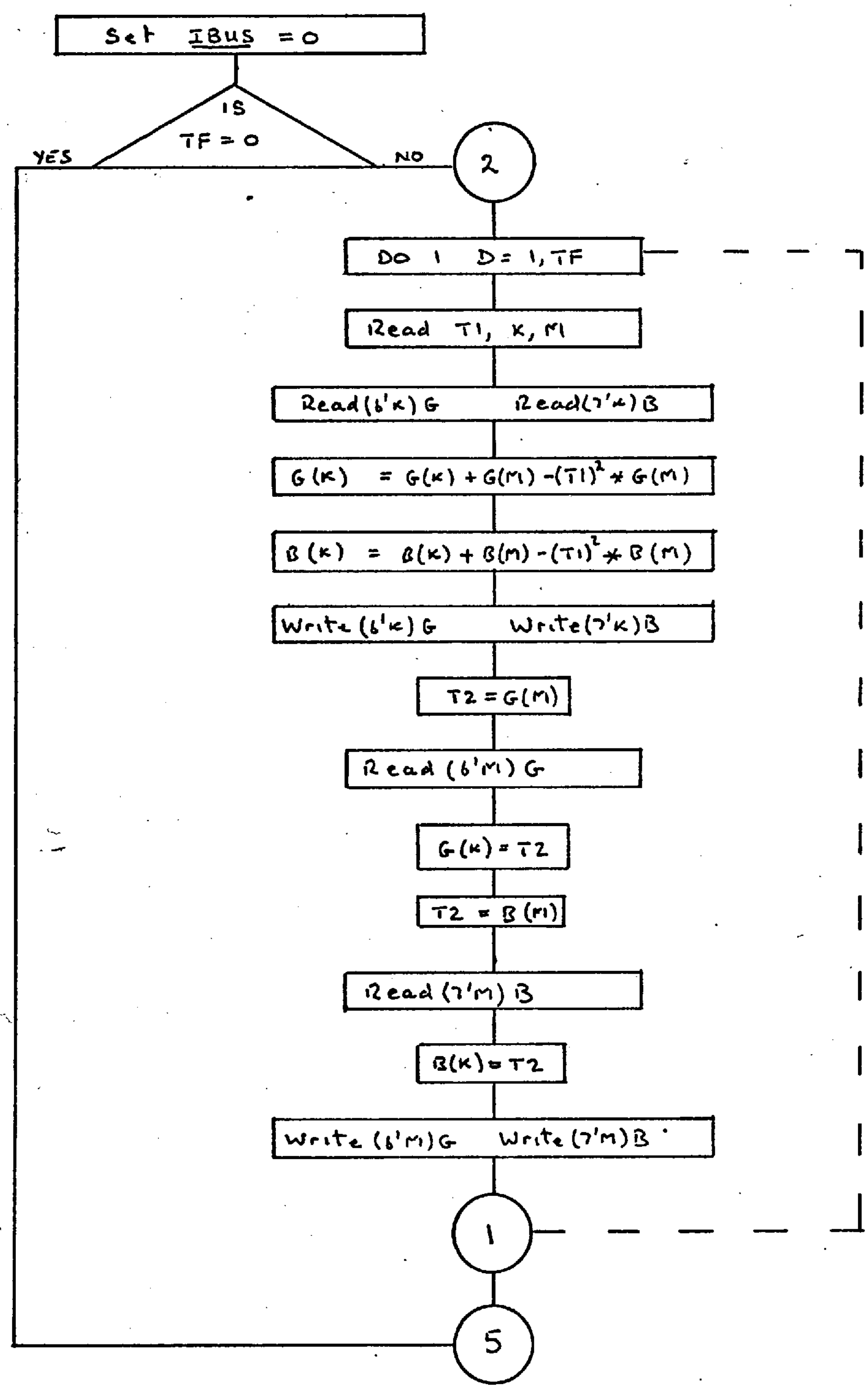


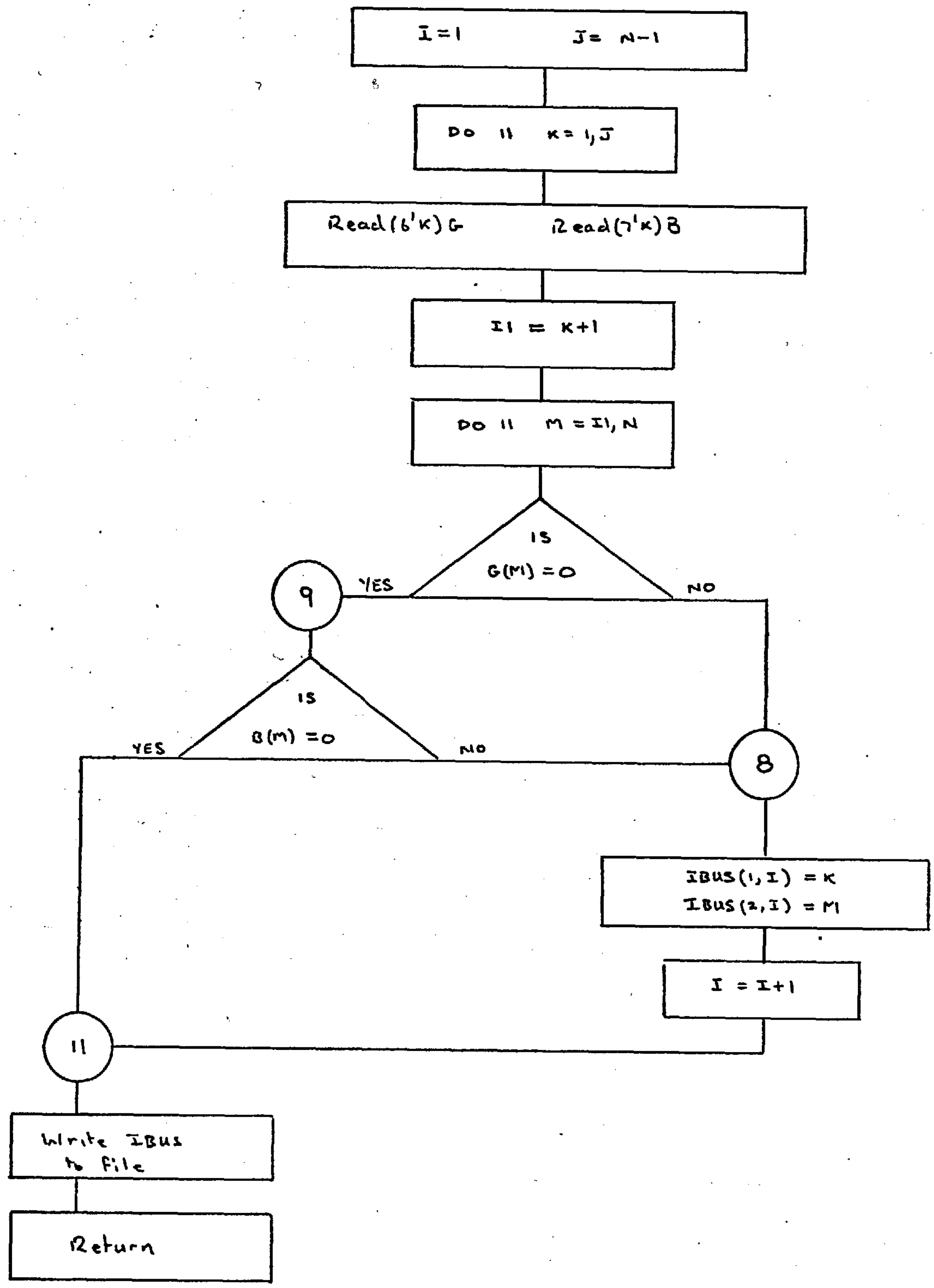
6.8 Subroutine PSA6

Subroutine PSA6 is used to modify the equivalent transformer impedances so as to allow for transformers which are adjusted to an off-nominal tap ratio position. This is accomplished by replacing the existing branch impedance value by the same value multiplied by the square of the per unit turns-ratio.

A secondary function also performed by this routine, is the initialisation of a second branch location matrix IBUS. A second branch location matrix is required, because subsequent routines can modify this matrix, by deleting the locations of branches that have been open circuited by the subroutine BREAK.

Subroutine PSA6 - Flow Chart




I=1 J= N-1
DO 11 K= 1,J
Read(6'K)G Read(7'K)B
I1 = K+1
DO 11 M = I1,N
IS G(M) = O
YES
NO
9
IS B(M) = O
YES
NO
8
IBUS(1,I) = K
IBUS(2,I) = M
I = I+1
11
Write IBUS to File
Return

6.9 Subroutine PSA7

6.9.1 General

The primary purpose of this subroutine is to solve the positive sequence network equations, it also performs several secondary functions - as indicated by the macro flow chart - such as data preparation and modification for subsequently called subroutines.

As previously stated, see 4.5.1, the program was designed to iterate from the positive, negative and zero sequence matrices which are accessed one row at a time. However, it has been possible to accomodate a triangularised admittance matrix in the core store, capable of accepting systems of up to 40 nodes. This eliminates one disc to core transfer for every iteration, saving ten milli-seconds for each iteration. Therefore, the first section of PSA7 obtains relevant from the data files and compiles the triangularised admittance matrix.

6.9.2 Network Modifications

Subroutine PSA7, as can be seen from the flow chart of the master routine, is re-entered several times during the course of a protection study. It is possible that before returning to PSA7 some of the system nodes have been isolated owing to network modifications which are initiated by circuit breaker operation. Therefore, a search for isolated nodes is made before the iteration procedure commences.

6.9.3 Short Circuit Studies

The operating system, and method of data storage permits a symmetrical three phase fault analysis to be made at up to 20 nodes on any given system for one data input operation, see 6.9.3. As this subroutine is re-entered each time a new three phase fault study is required, it is therefore necessary for it to perform the following ancillary functions:

1. Check the master short circuit matrix NODESC to see if a further study is required.

2. If a further study is required, to set the nodal designation tag ND7(k) to zero.

3. Set the voltage at the faulted node to zero.

4. When the solution voltage vector has been obtained, to reset the nodal designation tag to its original value.

6.9.4. The Iterative Procedure

The iterative procedure, detailed in chapter 2, obtains the solution vector to the equations 2.5, 2.6 and 2.5.1, 2.9.1. The Pade acceleration, see 2.4.1, is applied after the first fifteen iterations, which allows each voltage time to settle down and attain a constant direction of change; either increasing or decreasing. Thereafter, the acceleration is applied every ten iterations.

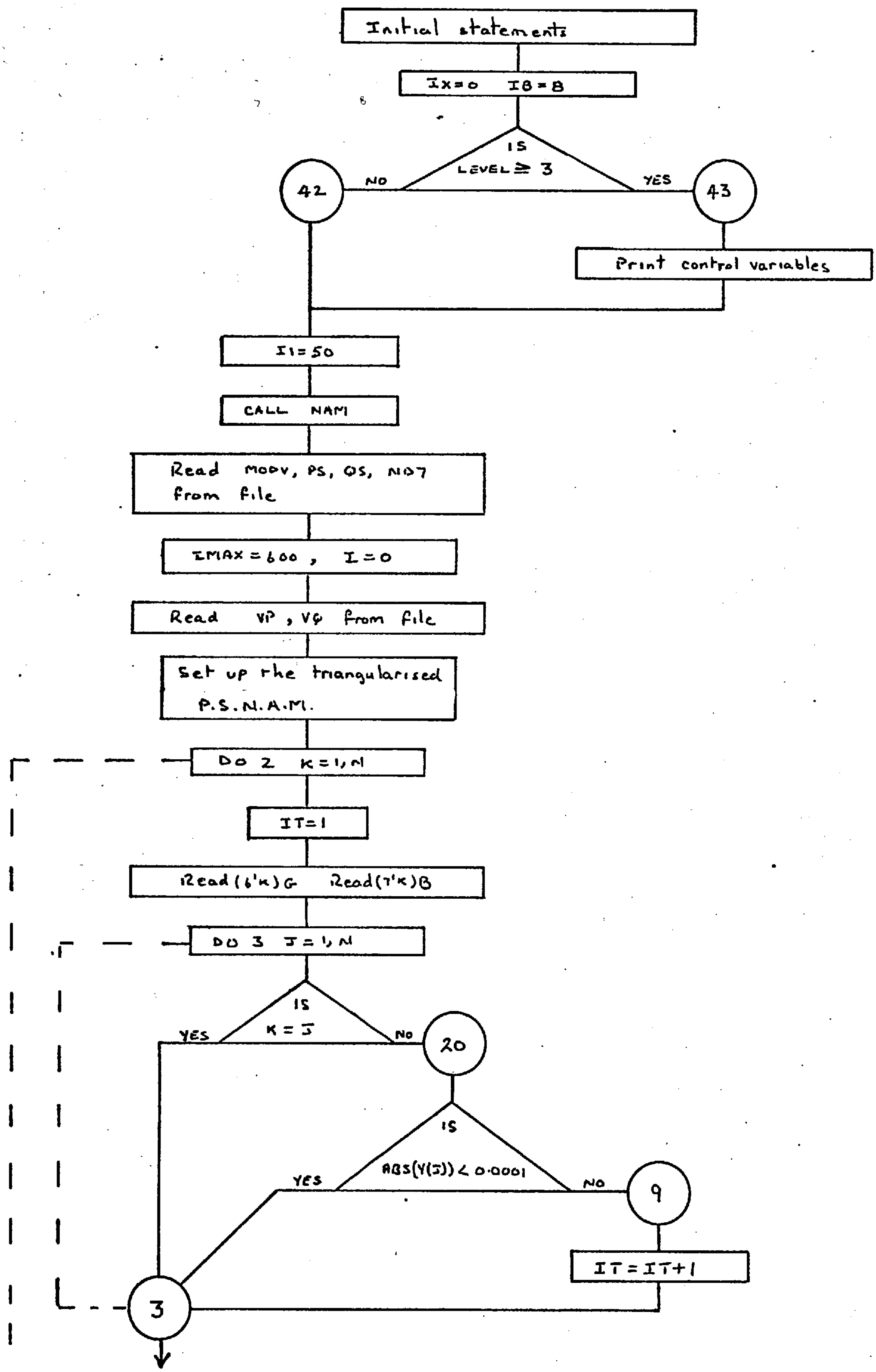
6.9.5 Machine EMF Calculation

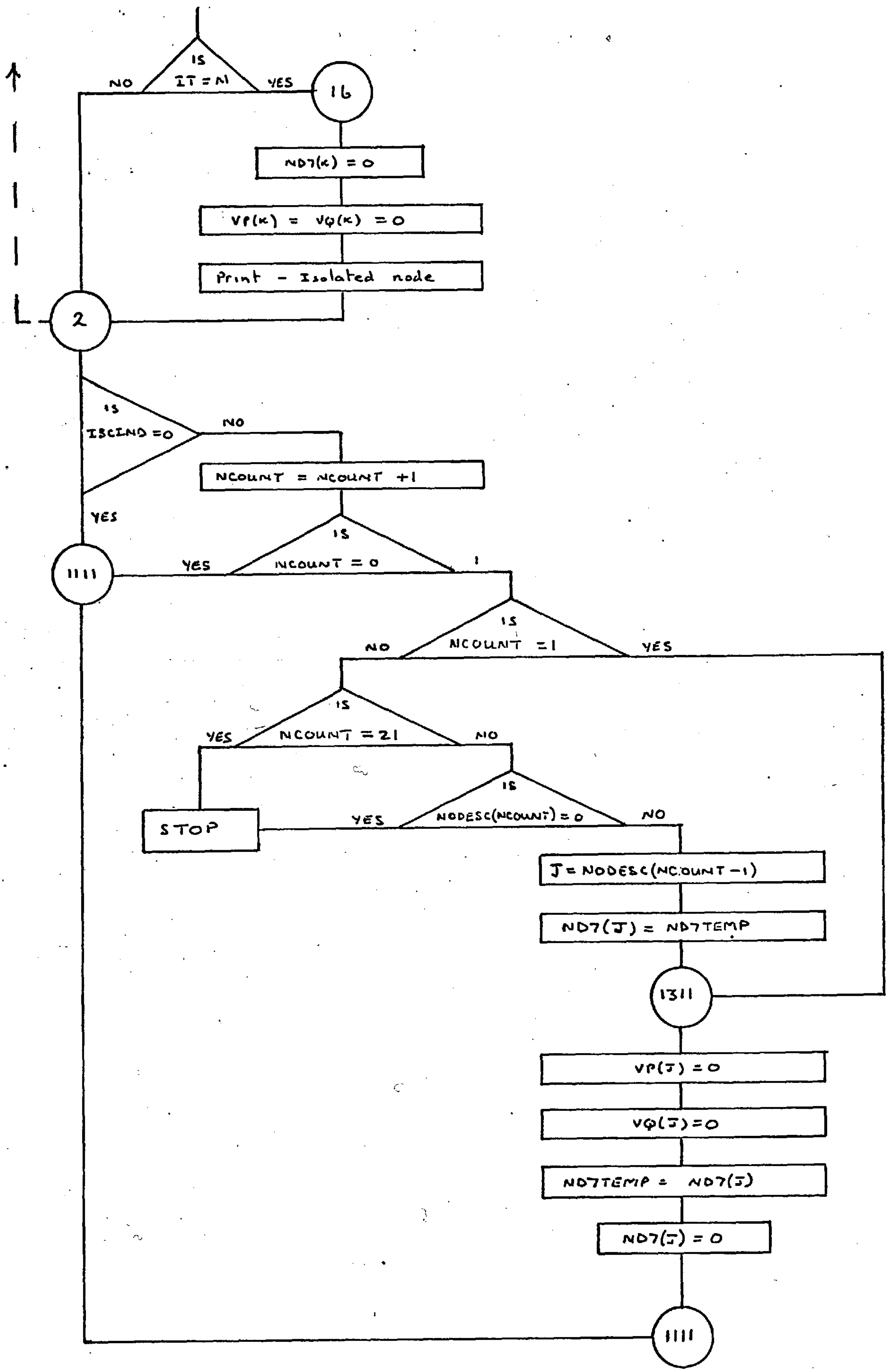
Prior to a fault study the machine emfs have to be evaluated, this calculation which follows the solution of the load flow equations, is initiated by specifying IEMF; see 5.5 (c).

6.9.6 Change of Nodal Designation

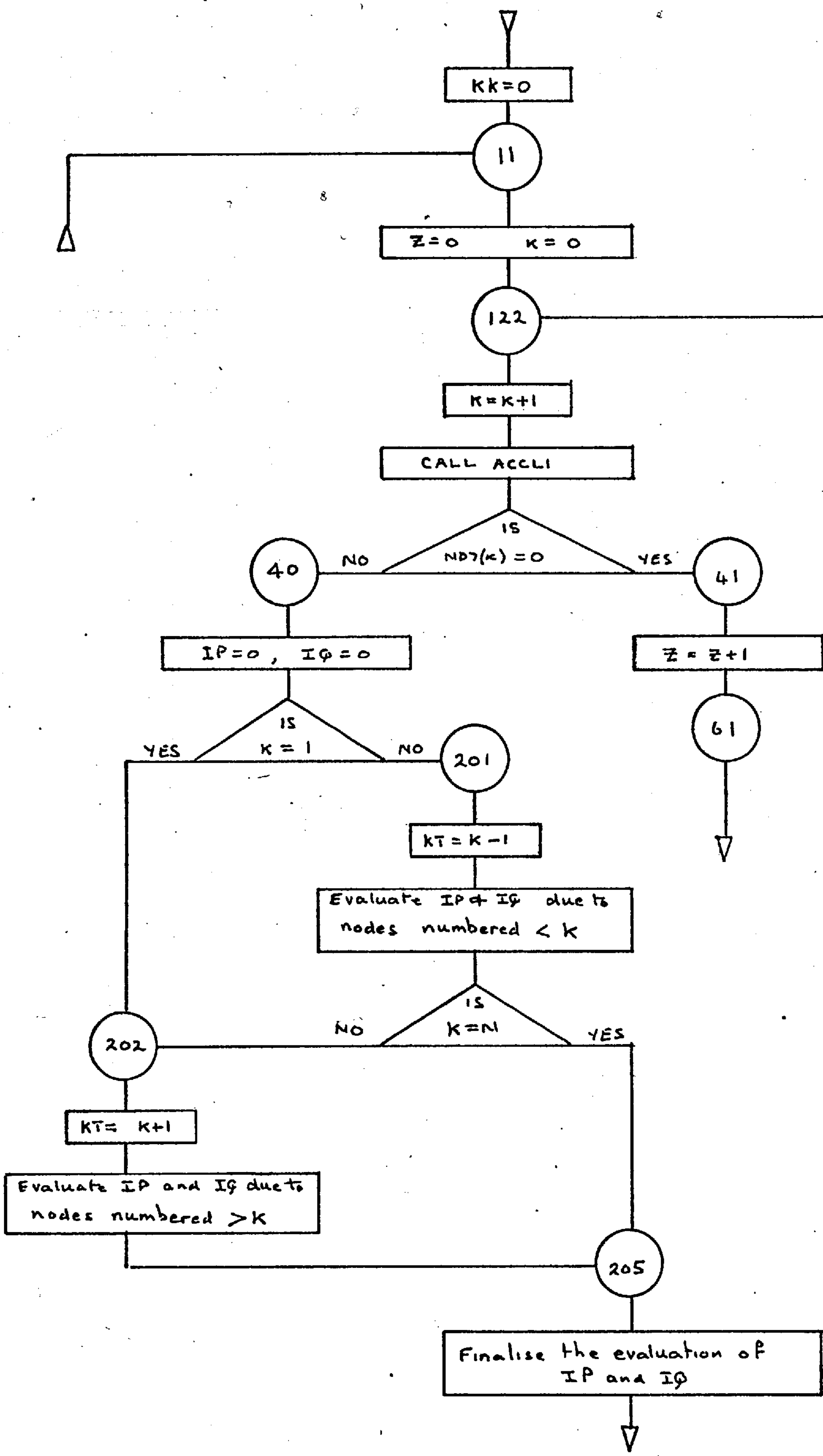
Finally, before the computed results are written to the data files, the nodal designation values for all generator nodes are changed from -1 to 1. Subroutine PSNS1 will subsequently change the generator nodes into constant impedance nodes, see 4.10 and 6.10.

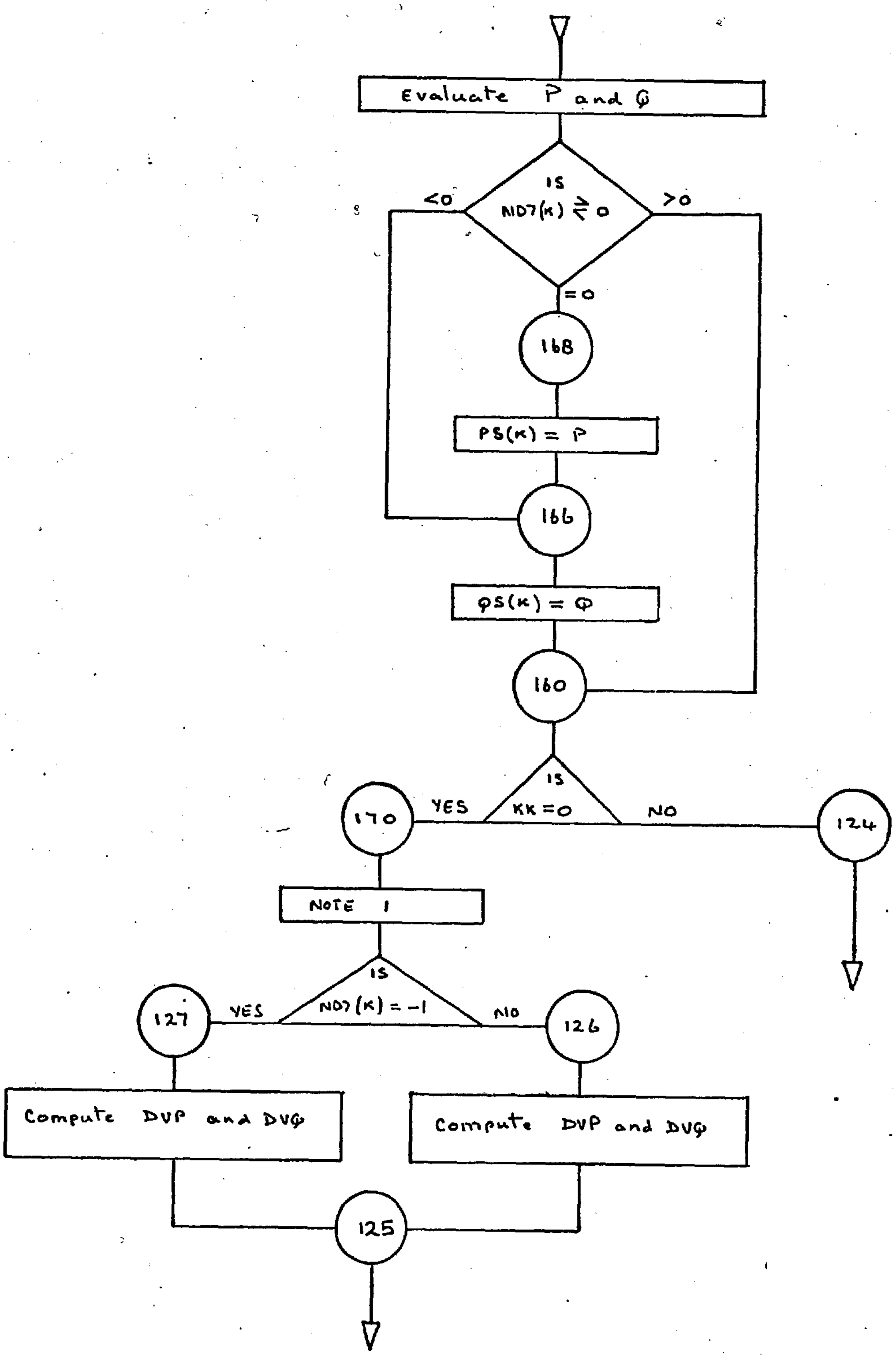
Subroutine PSA7 - Flow Chart



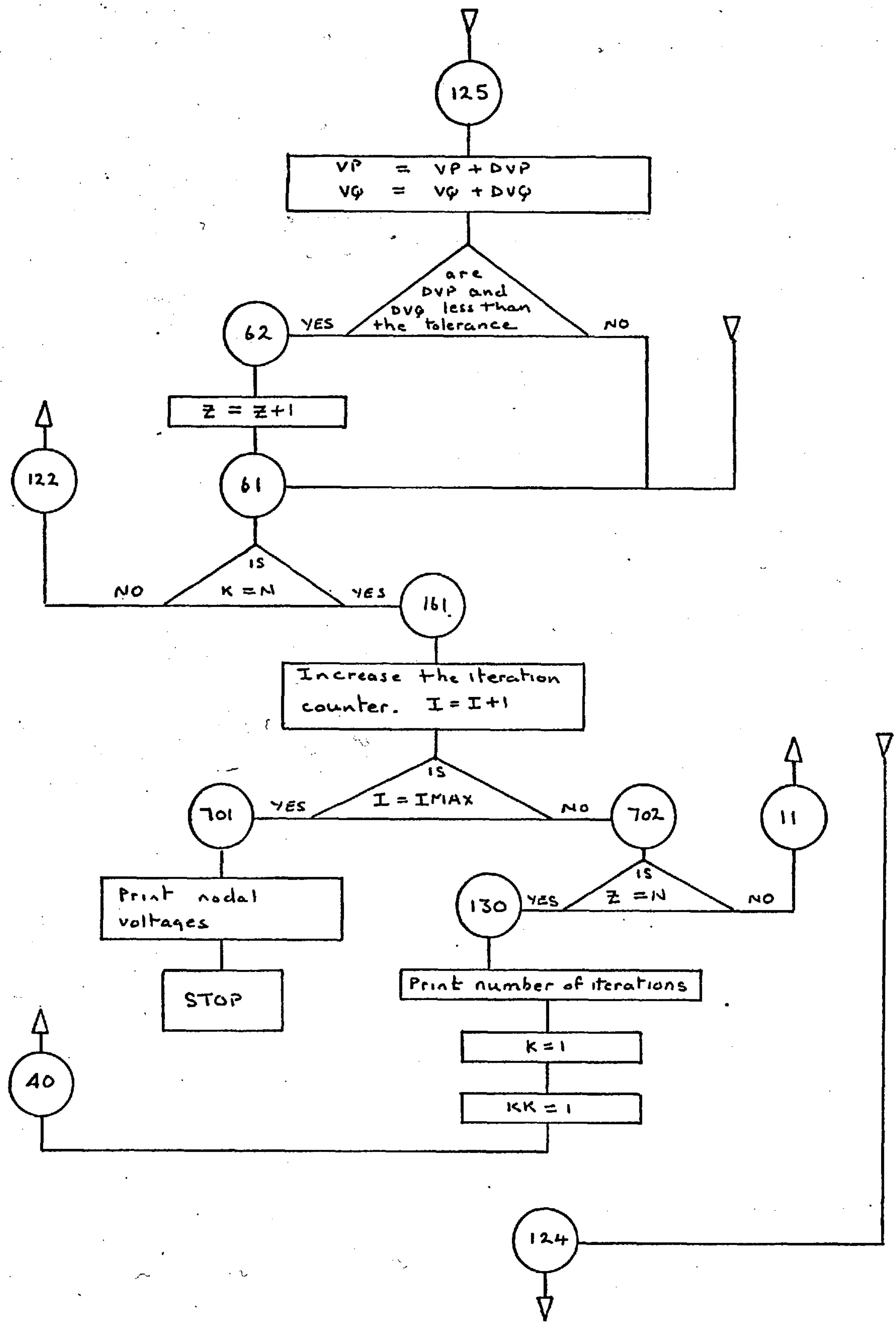


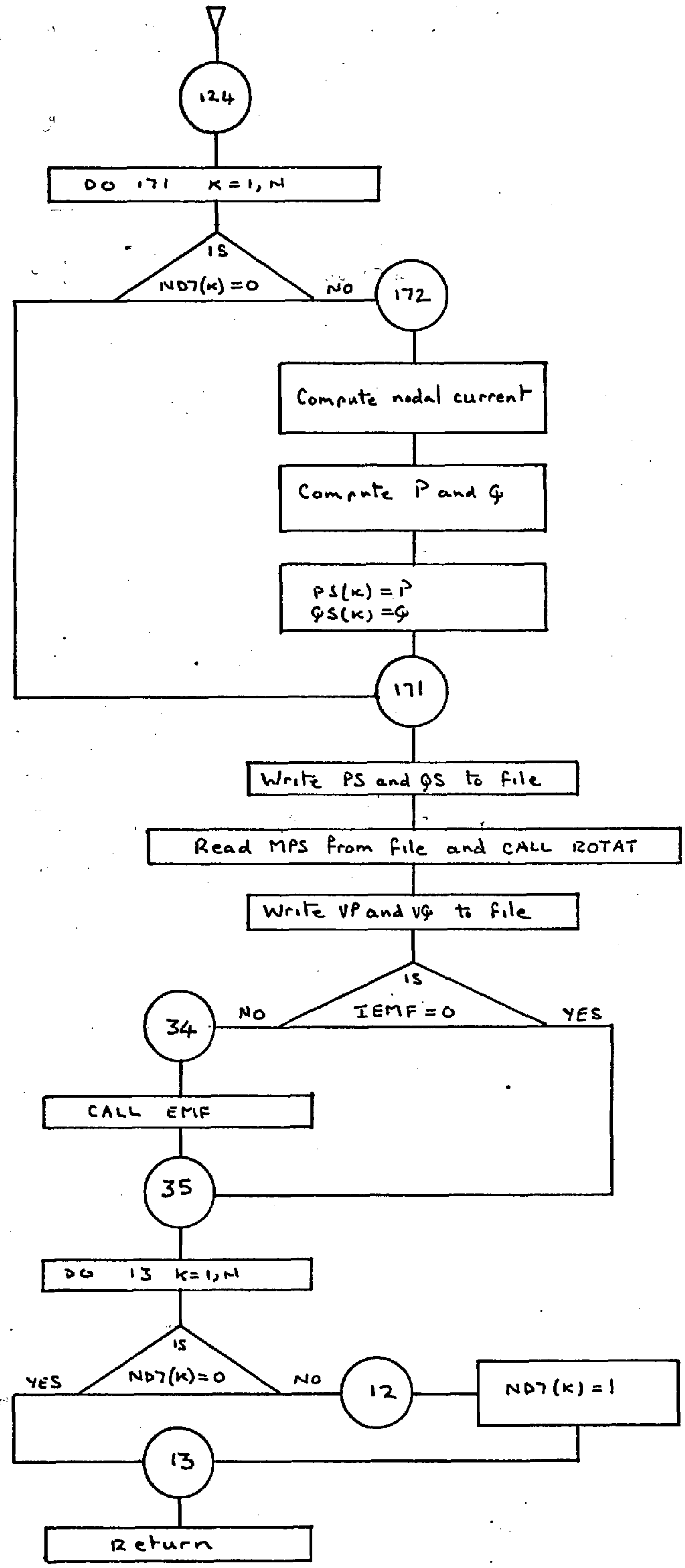
IS
IT = N
NO
YES
16
ND7(K) = 0
VP(K) = VQ(K) = 0
Print - Isolated node
2
IS
ISCIND = 0
NO
YES
NCOUNT = NCOUNT +1
IS
NCOUNT = 0
YES
1111
IS
NCOUNT = 1
NO
YES
IS
NCOUNT = 21
YES
NO
STOP
IS
NODESC(NCOUNT) = 0
YES
NO
J = NODESC(NCOUNT -1)
ND7(J) = ND7TEMP
1311
VP(J) = 0
VQ(J) = 0
ND7TEMP = ND7(J)
ND7(J) = 0
1111


KK=0
11
Z=0 K=0
122
K=K+1
CALL ACCLI
IS ND7(K)=0
NO
YES
40
41
IP=0 , IQ=0
Z=Z+1
61
IS K=1
YES
NO
201
KT=K-1
Evaluate IP & IQ due to nodes numbered < K
IS K=N
NO
YES
202
KT= K+1
Evaluate IP and IQ due to nodes numbered >K
205
Finalise the evaluation of IP and IQ



NOTE 1. This part of the calculation is common to both PV and PQ nodes.


125
VP = VP + DVP
VQ = VQ + DVQ
are DVP and DVQ less than the tolerance
YES
NO
62
Z = Z+1
122
61
IS K = N
NO
YES
161
Increase the iteration counter. I = I+1
IS I = IMAX
YES
NO
701
702
11
Print nodal voltages
STOP
130
IS Z = N
YES
NO
Print number of iterations
K = 1
KK = 1
40
124


124
DO 171 K=1,N
IS
ND7(K)=0
NO
172
Compute nodal current
Compute P and Q
PS(K)=P
QS(K)=Q
171
Write PS and QS to file
Read MPS from file and CALL ROTAT
Write VP and VQ to file
IS
IEMF=0
NO
YES
34
CALL EMF
35
DO 13 K=1,N
IS
ND7(K)=0
YES
NO
12
ND7(K)=1
13
Return

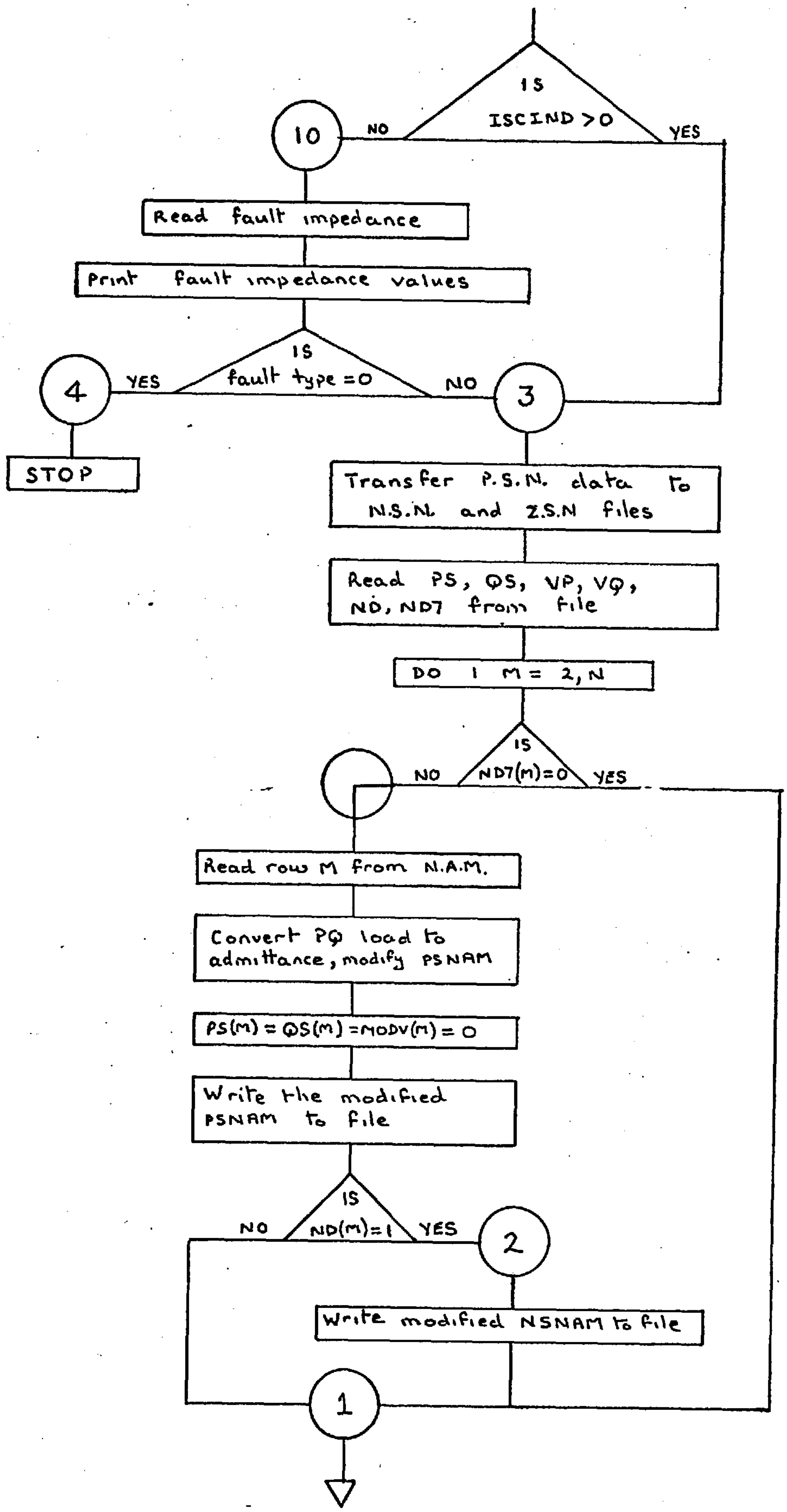
6.10 Subroutine PSNS1

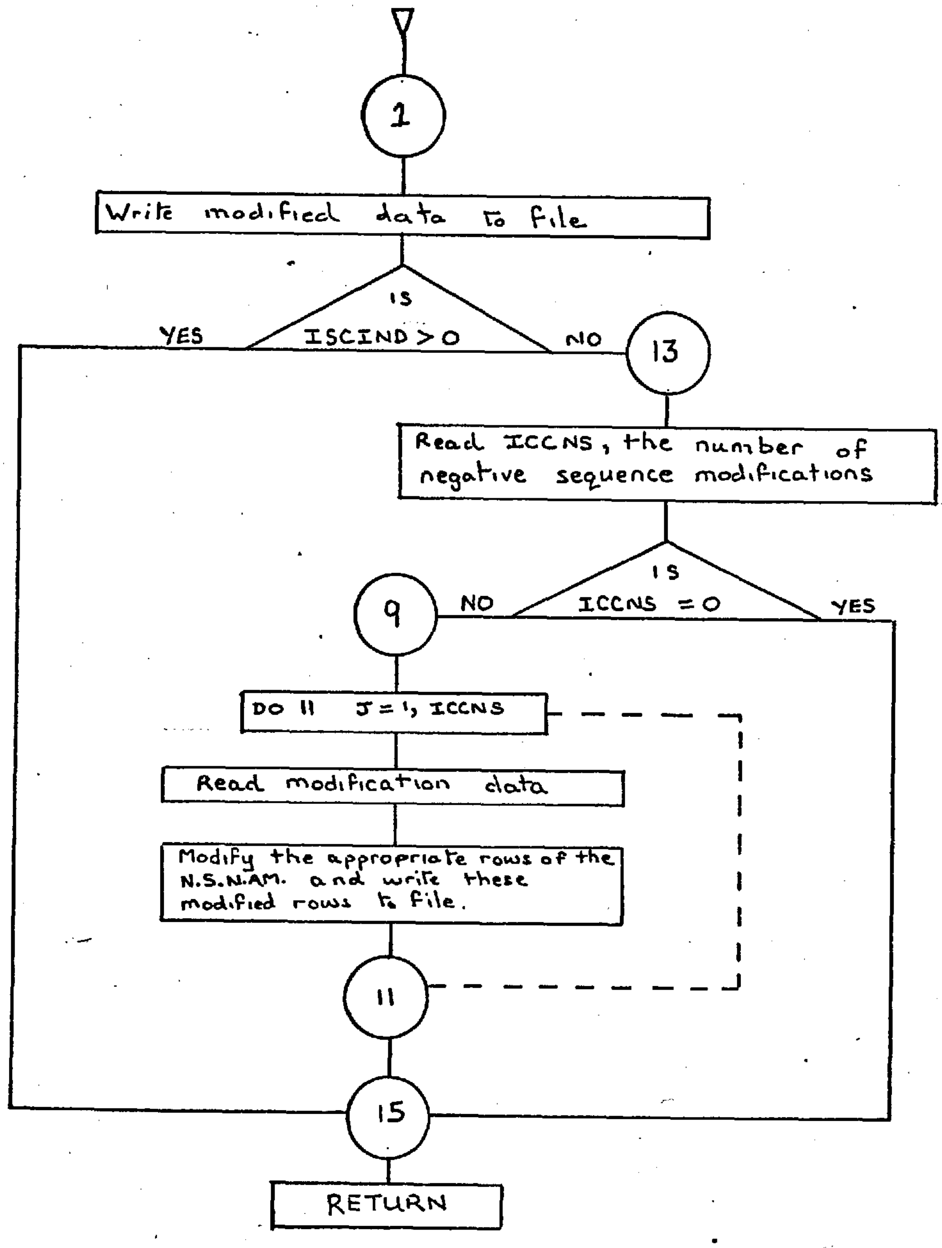
This routine and the one that follows, are designed to assemble and solve the system equations that relate to the negative sequence network. Subroutine PSNS1 assembles the negative sequence nodal admittance matrix from the P.S.N.A.M. data, incorporating any modifications as required. It also converts loads specified as P,Q loads into an equivalent constant impedance form.

6.11 Subroutine PSNS2

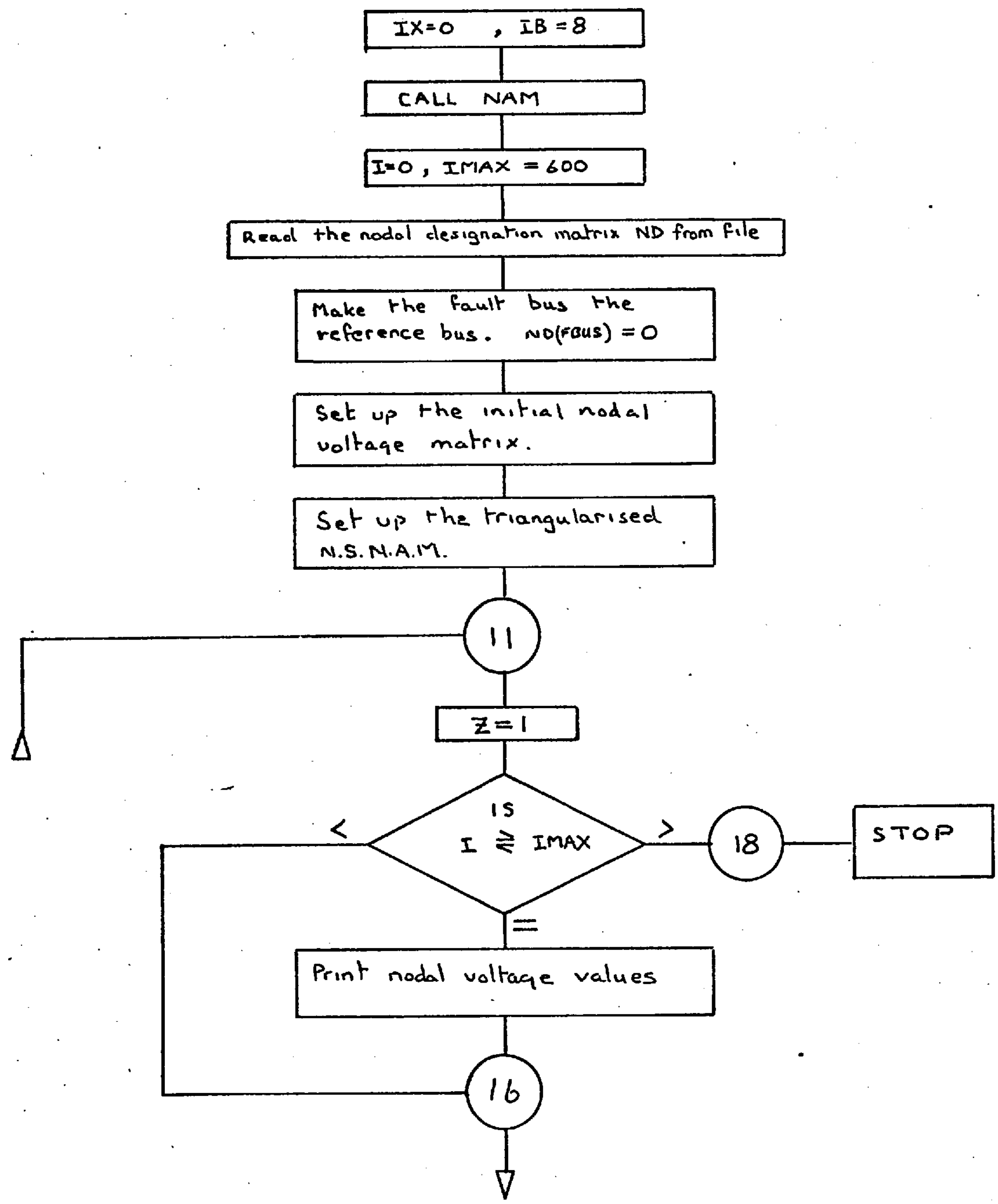
Subroutine PSNS2 is used to solve the N.S.N.A.M. equations that have been assembled by PSNS1. The method of solution is identical to that used by PSA7 for the P.S.N.A.M. equations. When the negative sequence nodal voltages have been computed, they are stored in the appropriate data files.

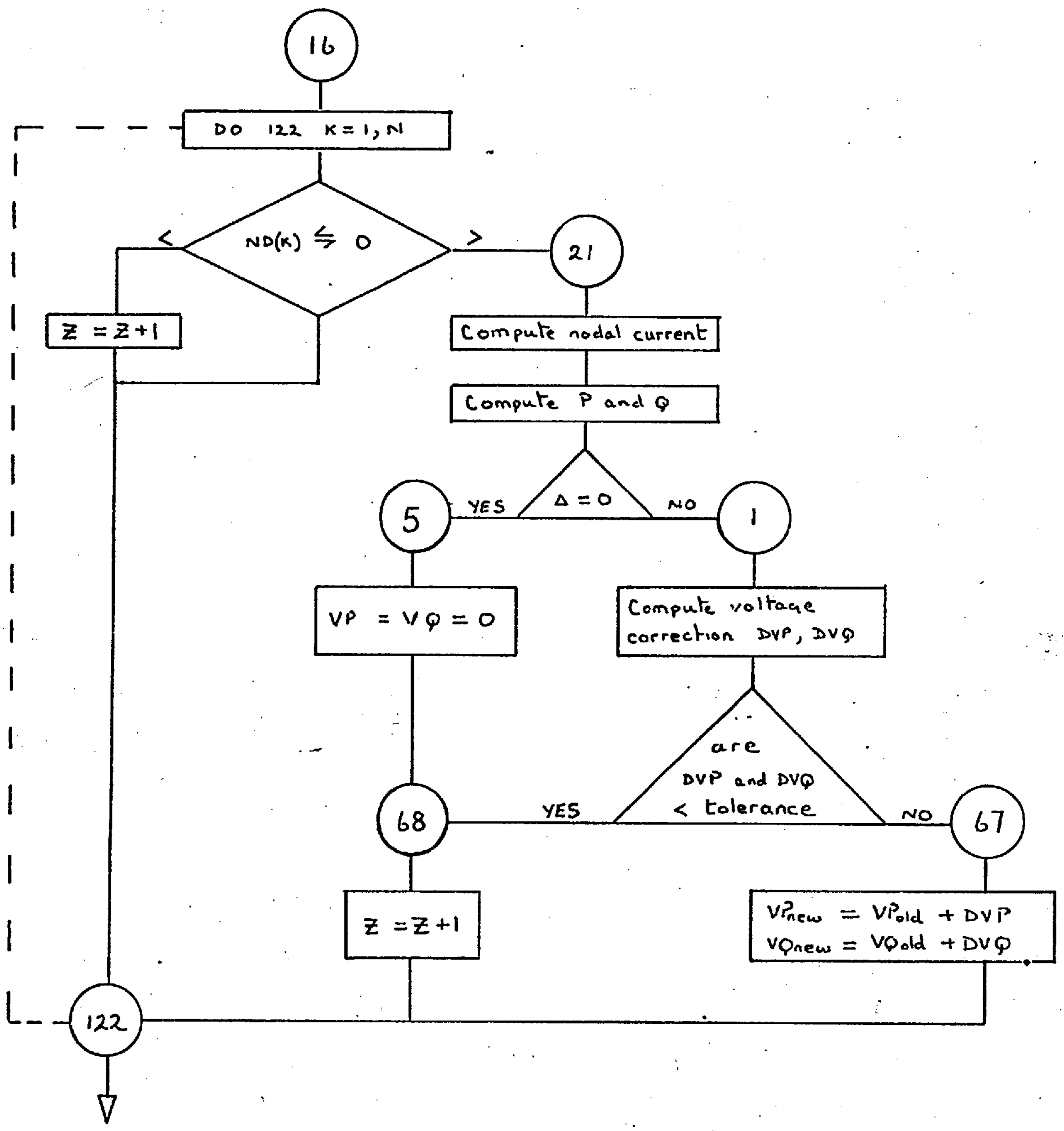
Subroutine PSNS1 - Flow Chart

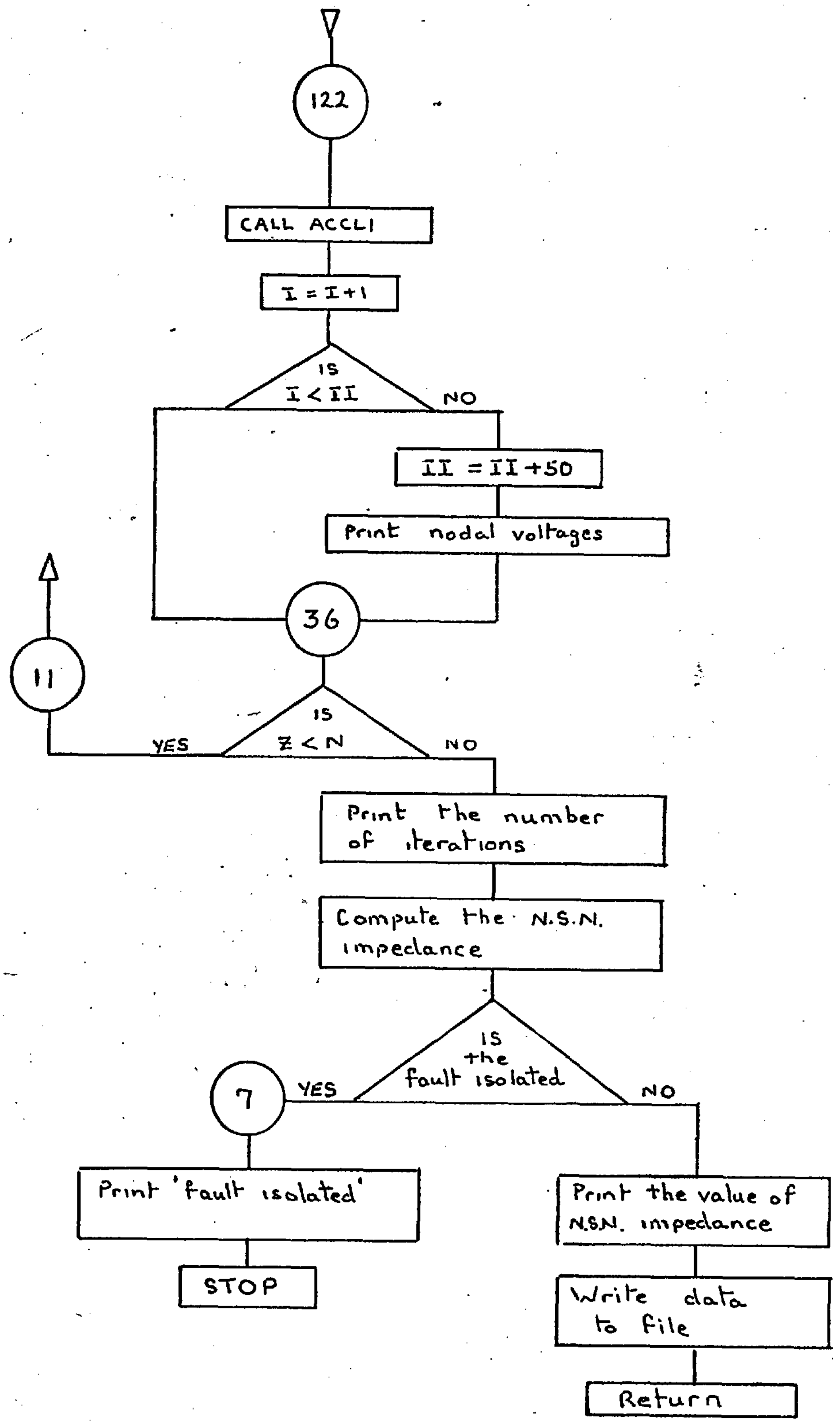



1
Write modified data to file
IS
ISCIND > O
YES
NO
13
Read ICCNS, the number of negative sequence modifications
IS
ICCNS = O
NO
YES
9
DO 11 J = 1, ICCNS
Read modification data
Modify the appropriate rows of the N.S.N.A.M. and write these modified rows to file.
11
15
RETURN

Subroutine PSNS2 - Flow Chart




16
DO 122 K=1,N
ND(K) ⋚ 0
<
>
21
Z = Z+1
Compute nodal current
Compute P and Q
Δ = 0
YES
NO
5
1
VP = VQ = 0
Compute voltage correction DVP, DVQ
are DVP and DVQ < tolerance
YES
NO
68
67
Z = Z+1
VPnew = VPold + DVP
VQnew = VQold + DVQ
122


122
CALL ACCLI
I = I+1
IS I < II
NO
II = II +50
Print nodal voltages
36
11
IS Z < N
YES
NO
Print the number of iterations
Compute the N.S.N. impedance
IS the fault isolated
YES
NO
7
Print 'fault isolated'
STOP
Print the value of N.S.N. impedance
Write data to file
Return

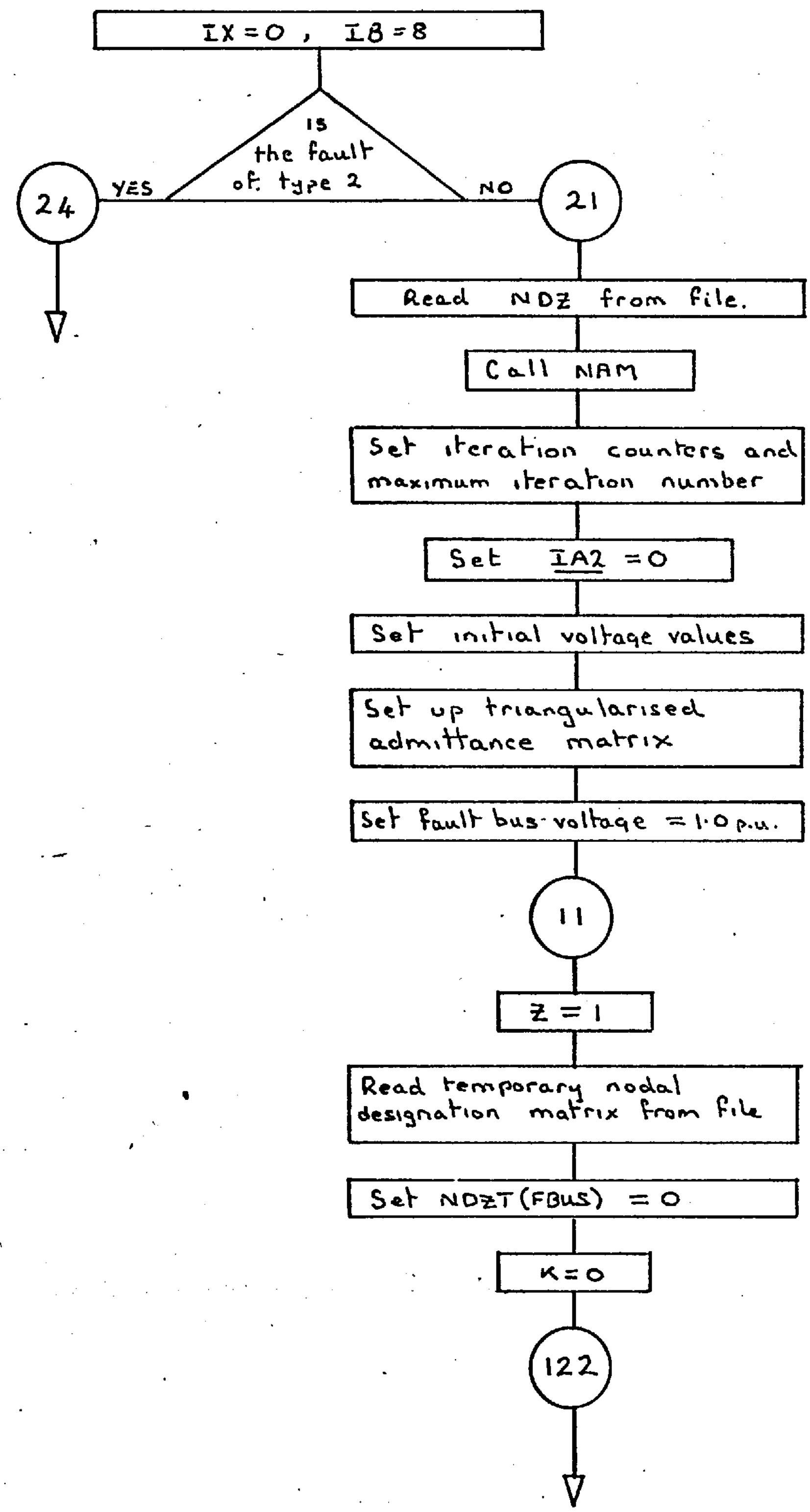
6.12 Subroutine PSAZS

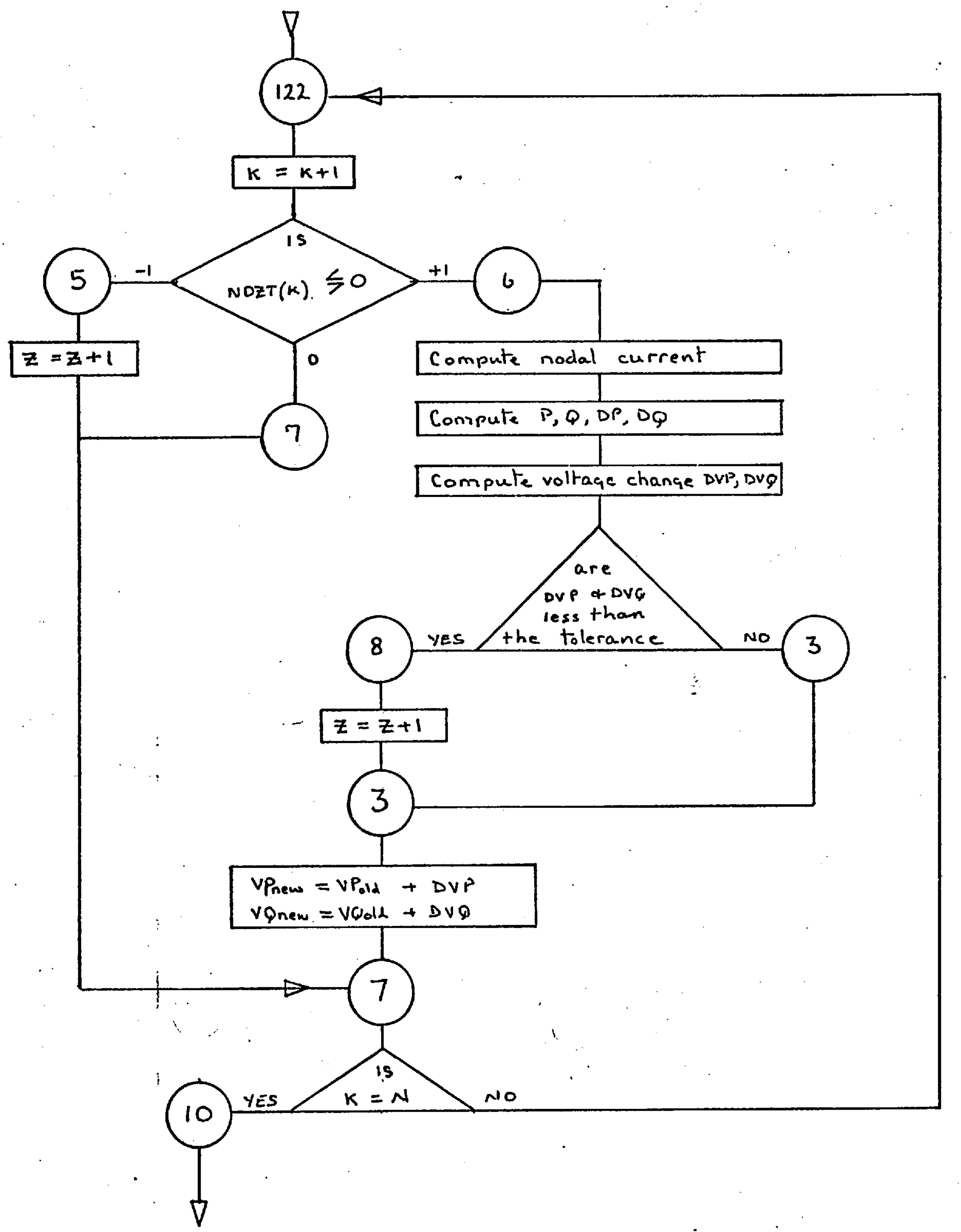
This subroutine corresponds to PSNS1, and is an equivalent zero sequence routine. PSAZS completes the assembly of the Z.S.N.A.M. (initiated by PSNS1) by including modifications for the transformer winding connections, as described in chapter 2. A check is made by this subroutine to determine if any of the network modifications have isolated any of the nodes from the remainder of the system. If any isolated nodes are detected, the starting value of the nodal voltage is changed to zero, and the nodal designation tag is set equal to zero; the isolated node is therefore removed from the iteration procedure.

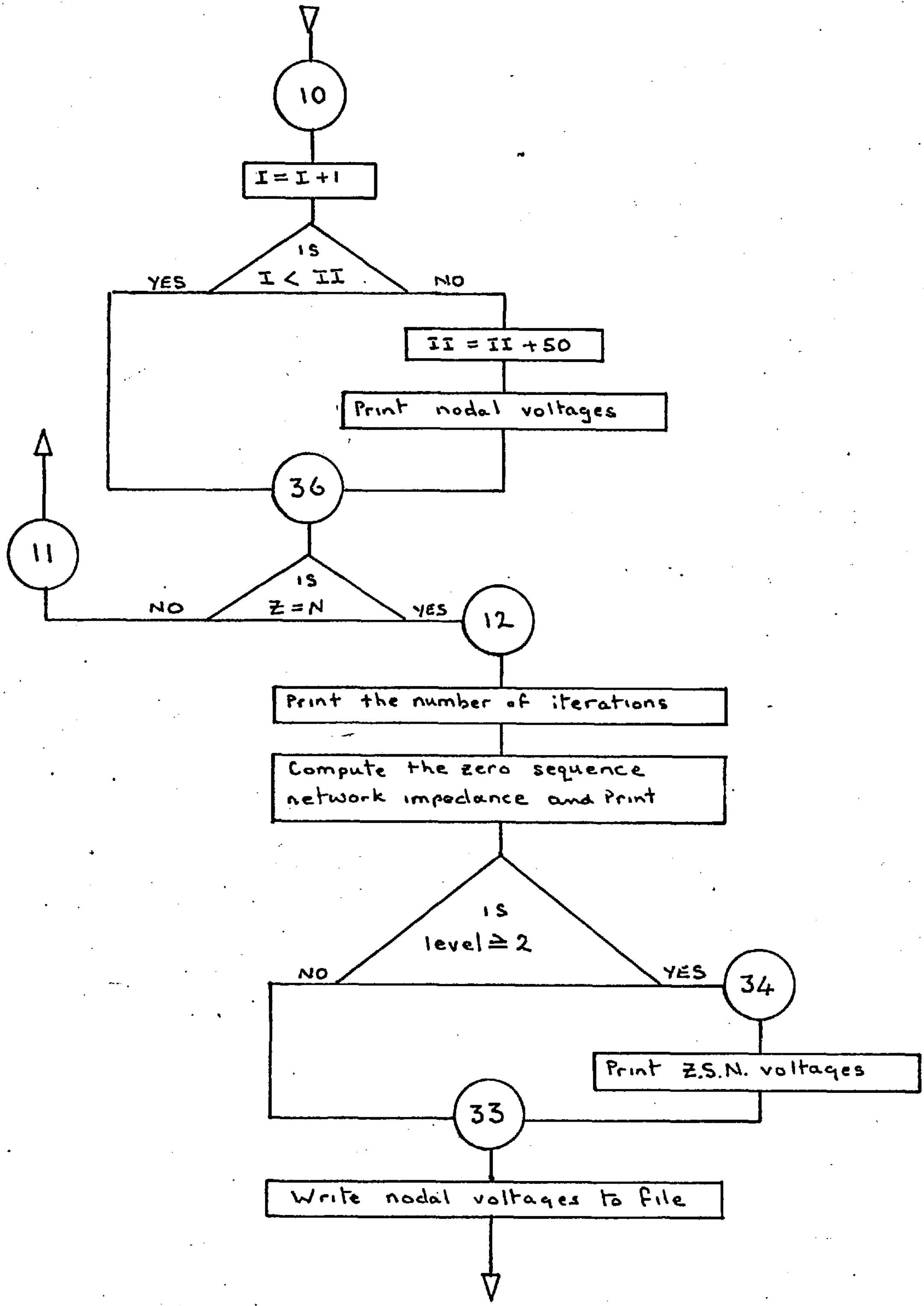
6.13 Subroutine ZSSOL

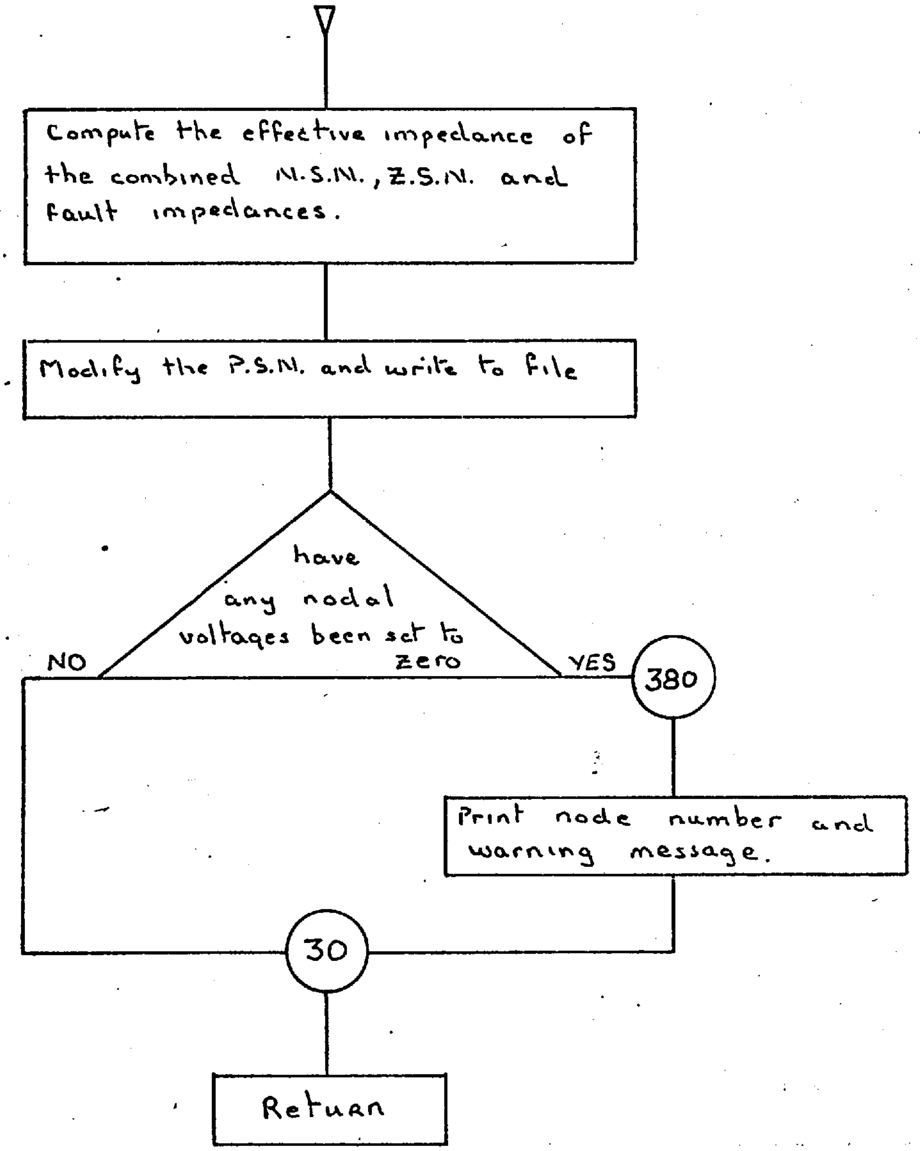
The main function of this subroutine is the solution of the Z.S.N.A.M. equations which have been assembled by PSAZS. The same method is used, as by PSA7 and PSNS2. When the nodal voltages have been computed they are stored in the data files. ZSSOL then evaluates the impedance of the Z.S.N. as seen from the fault point, combines this value with the corresponding value for the N.S.N. and then modifies the P.S.N. appropriately; as described in chapter 2.

Subroutine ZSSOL.   Flow Chart




122
K = K+1
IS
NDZT(K) ≶ 0
-1
+1
0
5
6
Z = Z+1
Compute nodal current
Compute P, Q, DP, DQ
Compute voltage change DVP, DVQ
7
are
DVP & DVQ
less than
the tolerance
YES
NO
8
3
Z = Z+1
3
VPnew = VPold + DVP
VQnew = VQold + DVQ
7
IS
K = N
YES
NO
10


10
I = I + 1
IS
I < II
YES
NO
II = II + 50
Print nodal voltages
36
11
IS
Z = N
NO
YES
12
Print the number of iterations
Compute the zero sequence network impedance and Print
IS
level ≥ 2
NO
YES
34
Print Z.S.N. voltages
33
Write nodal voltages to file


Compute the effective impedance of the combined N.S.N., Z.S.N. and fault impedances.
Modify the P.S.N. and write to file
have any nodal voltages been set to zero
NO
YES
380
Print node number and warning message.
30
Return

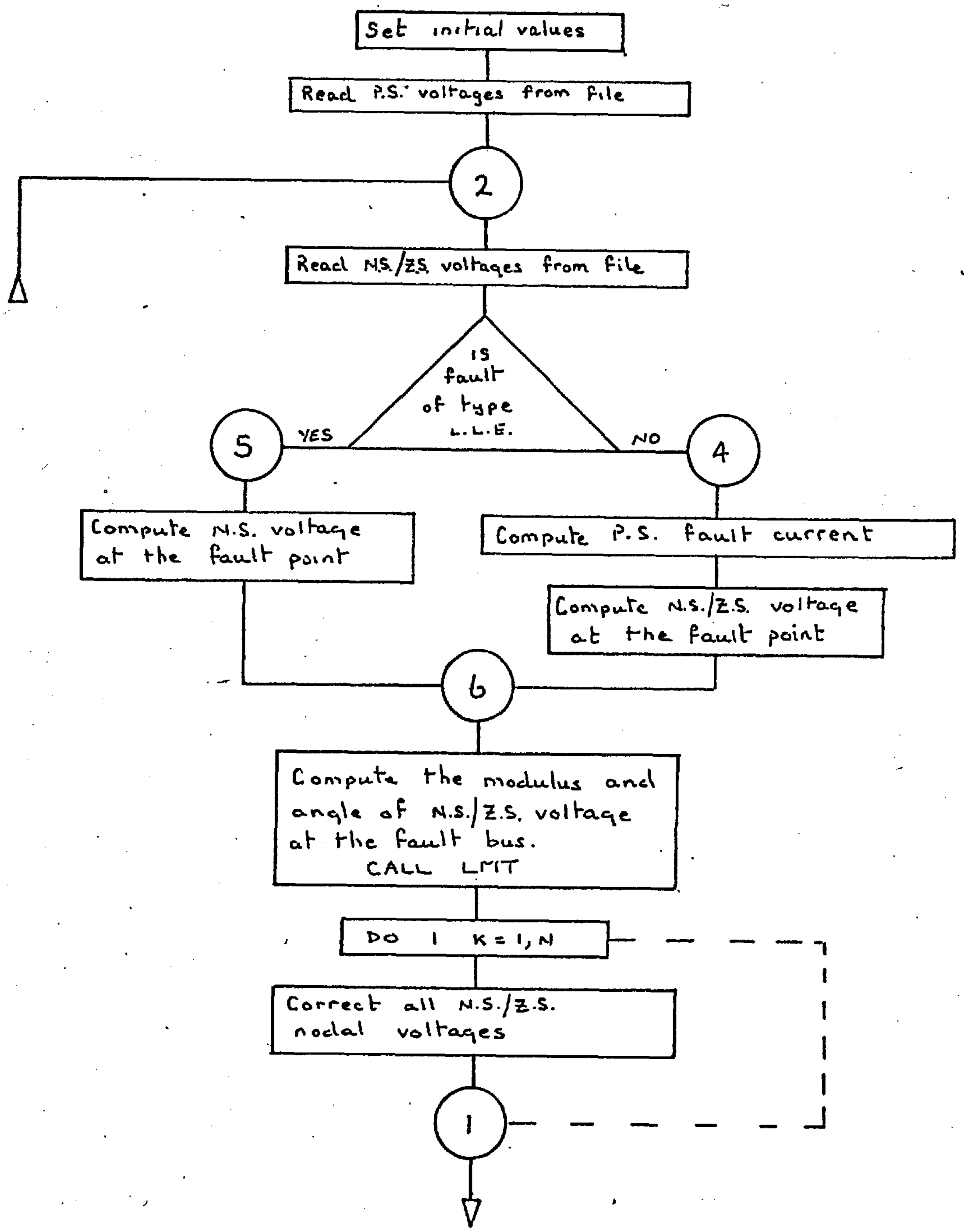
6.14 Subroutine VFCTR

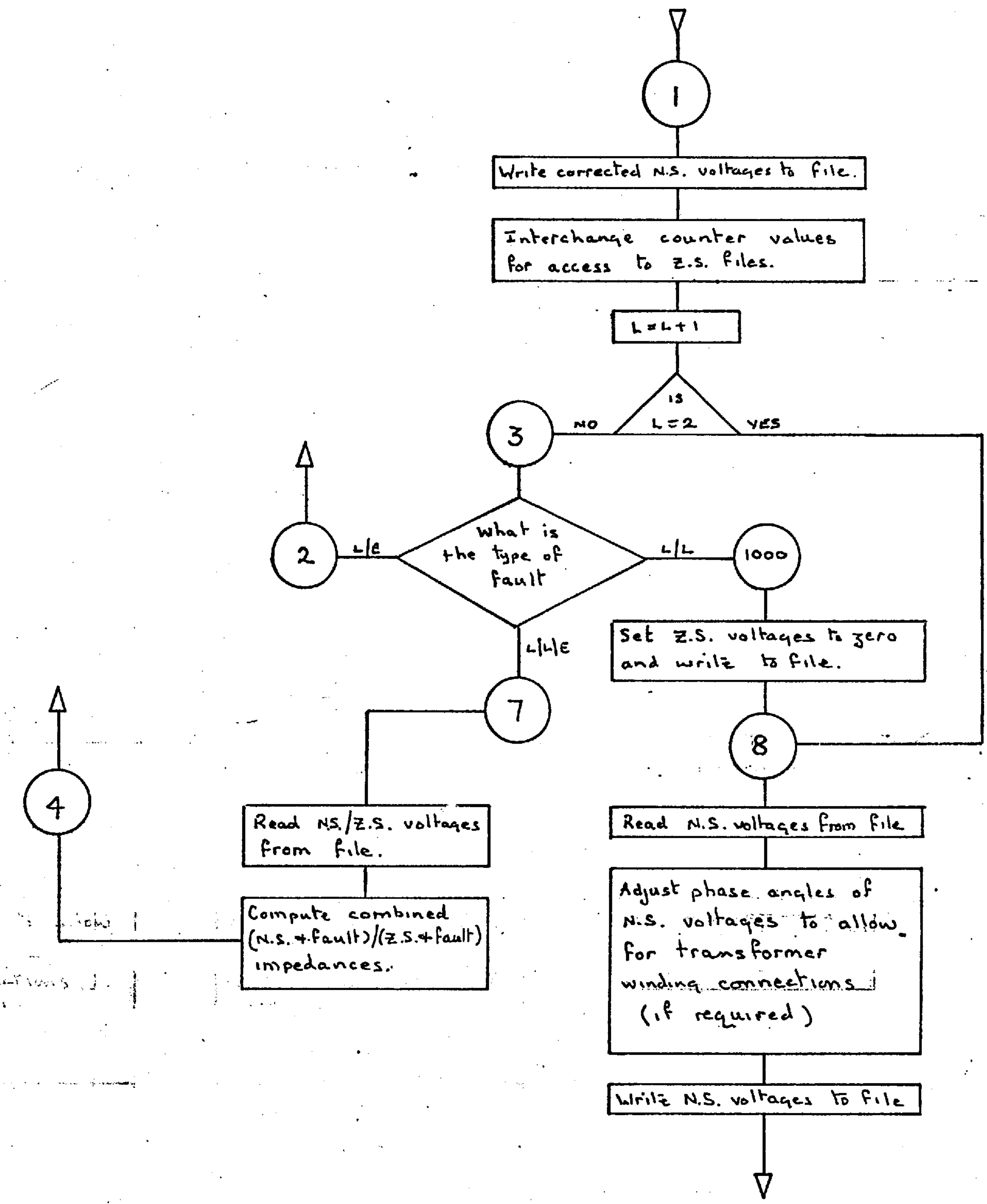
This subroutine computes the actual negative and zero sequence voltages at the fault point, these values are the scaling factors for the voltages computed by PSNS2 and ZSSOL respectively. After scaling, the negative and zero sequence voltages are stored in the data files.

6.15 Subroutine VCOMP

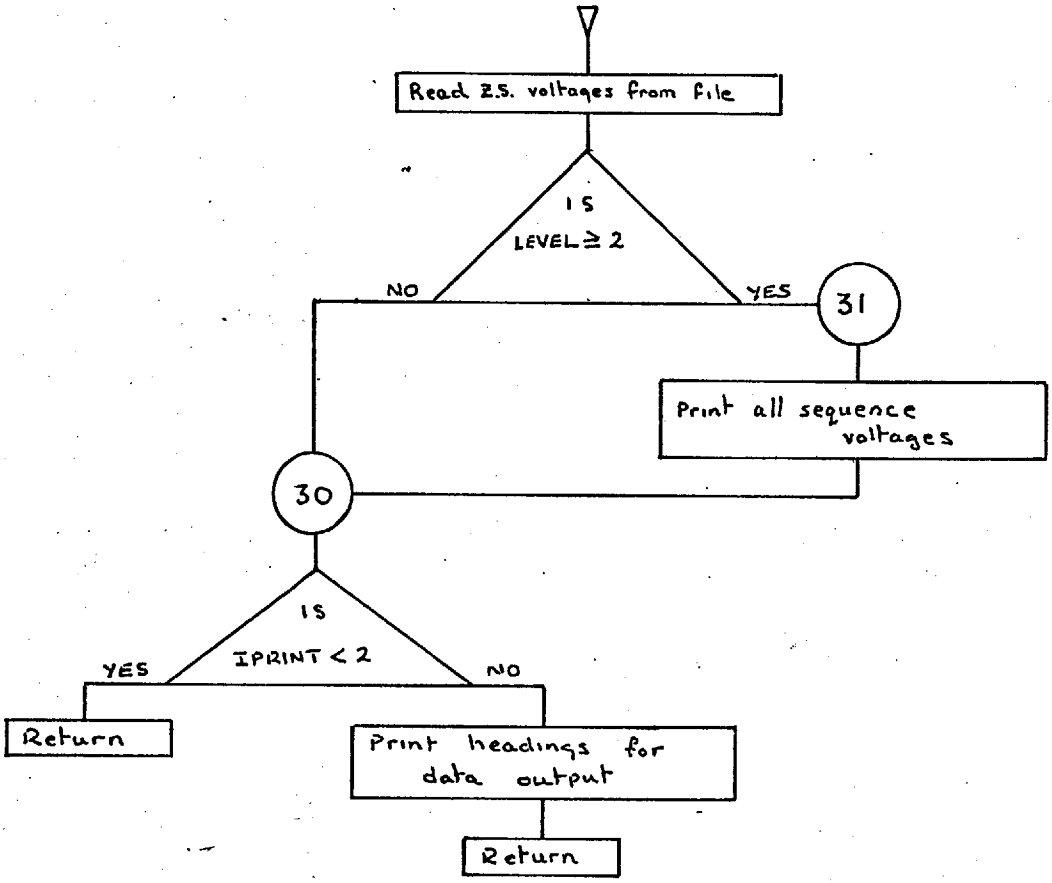
VCOMP reads the positive, negative and zero sequence voltages from the data files, and computes the three nodal phase voltages. For an asymmetric fault analysis this data is printed, however, if a protection study is being made this data is only printed if required; the control variable being IPRINT - see User Manual, appendix 1 .

Subroutine VFCTR. Flow Chart





1
Write corrected N.S. voltages to file.
Interchange counter values for access to Z.S. files.
L=L+1
IS L=2
NO
YES
3
What is the type of fault
L/E
2
L/L
1000
L/L/E
7
Set Z.S. voltages to zero and write to file.
8
4
Read NS./Z.S. voltages from file.
Compute combined (N.S.+fault)/(Z.S.+fault) impedances.
Read N.S. voltages from file
Adjust phase angles of N.S. voltages to allow for transformer winding connections (if required)
Write N.S. voltages to file

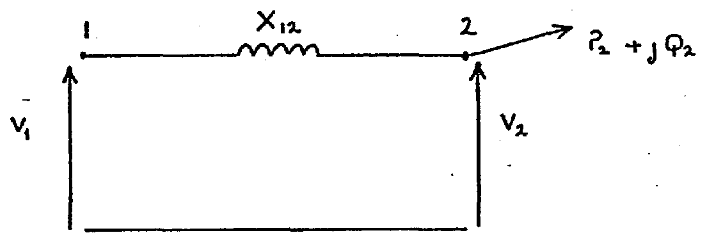

Read Z.S. voltages from file
IS
LEVEL ≥ 2
NO
YES
31
Print all sequence voltages
30
IS
IPRINT < 2
YES
NO
Return
Print headings for data output
Return

6.16 Subroutine EMF

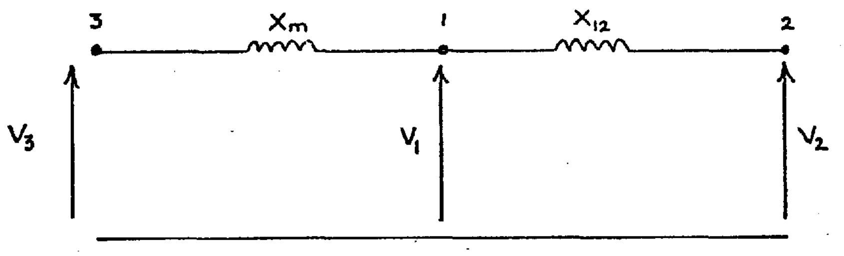
Subroutine EMF is used to compute the value of the emf acting behind the machine impedance. To do this it uses the results of the balanced load flow analysis, obtained from PSA7, and some additional machine data which is supplied as required.

For example:

A load flow is performed on the simple two node system shown below



where node 1 is the reference bus. If the machine at node 1 is to be represented by an emf acting behind an equivalent reactance, EMF reads the nodal location, the new node number and the value of reactance. The system is amended to



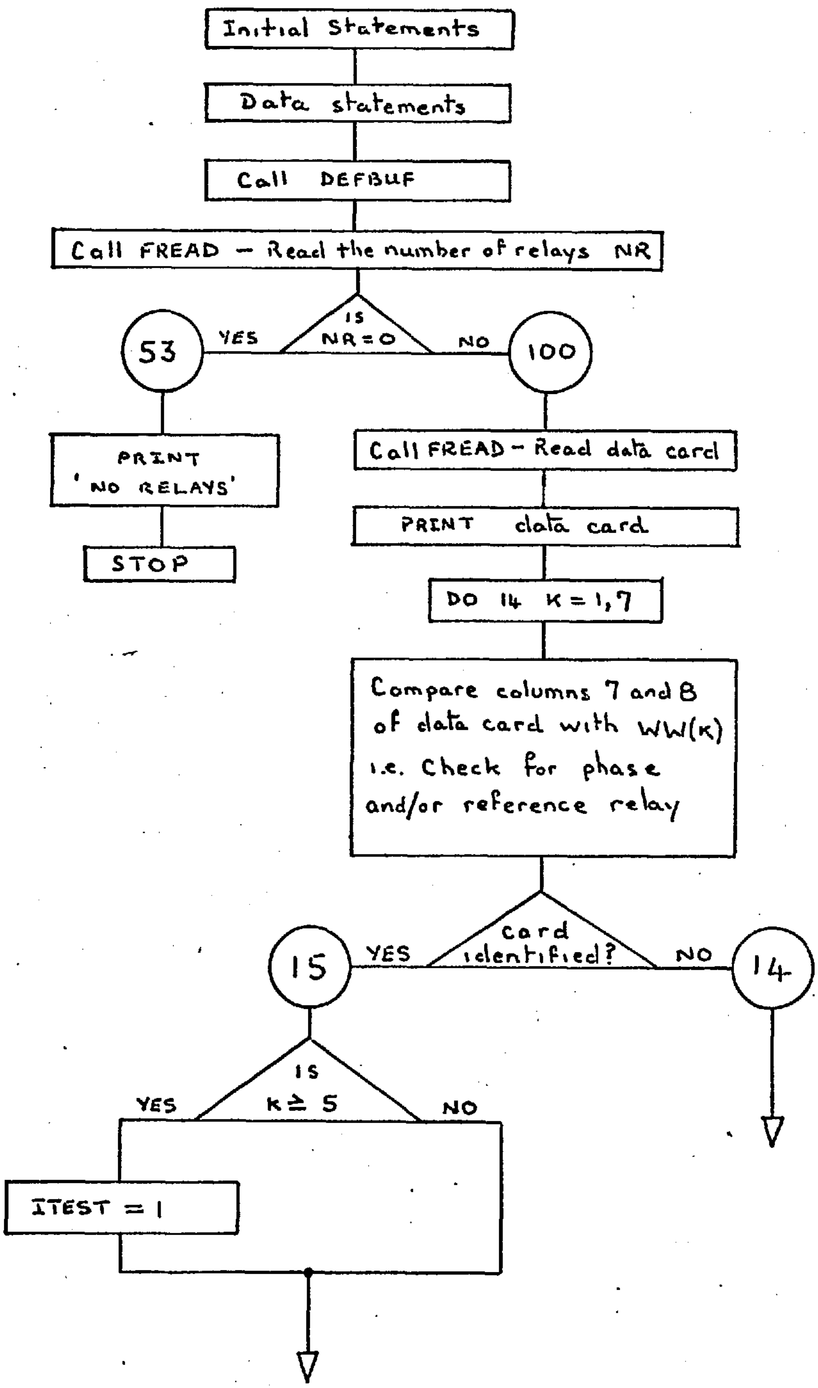
V_3 is then evaluated, using the results of the load flow analysis - I_{12} in this instance.

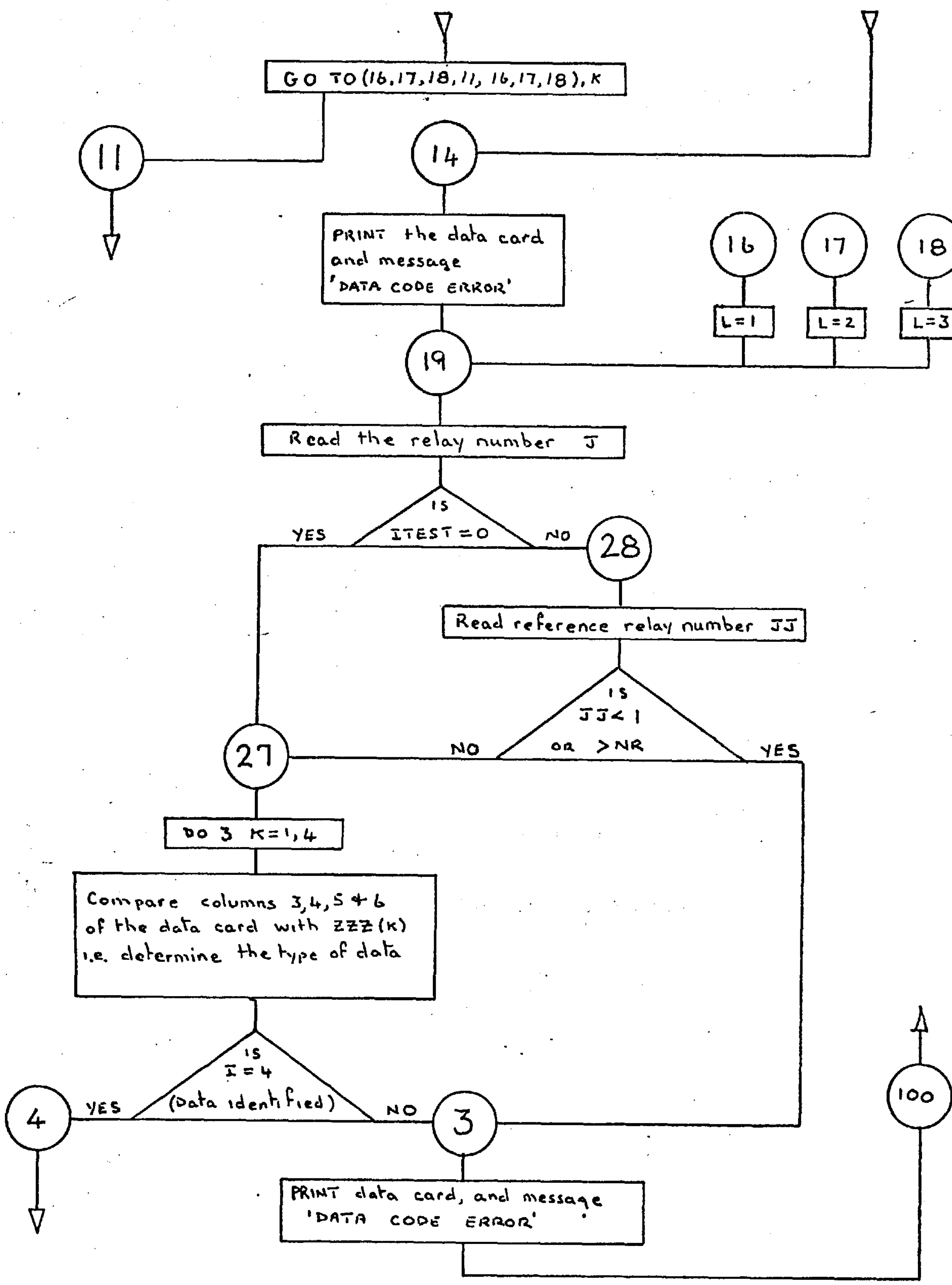
6.17 Subroutine RLDAT

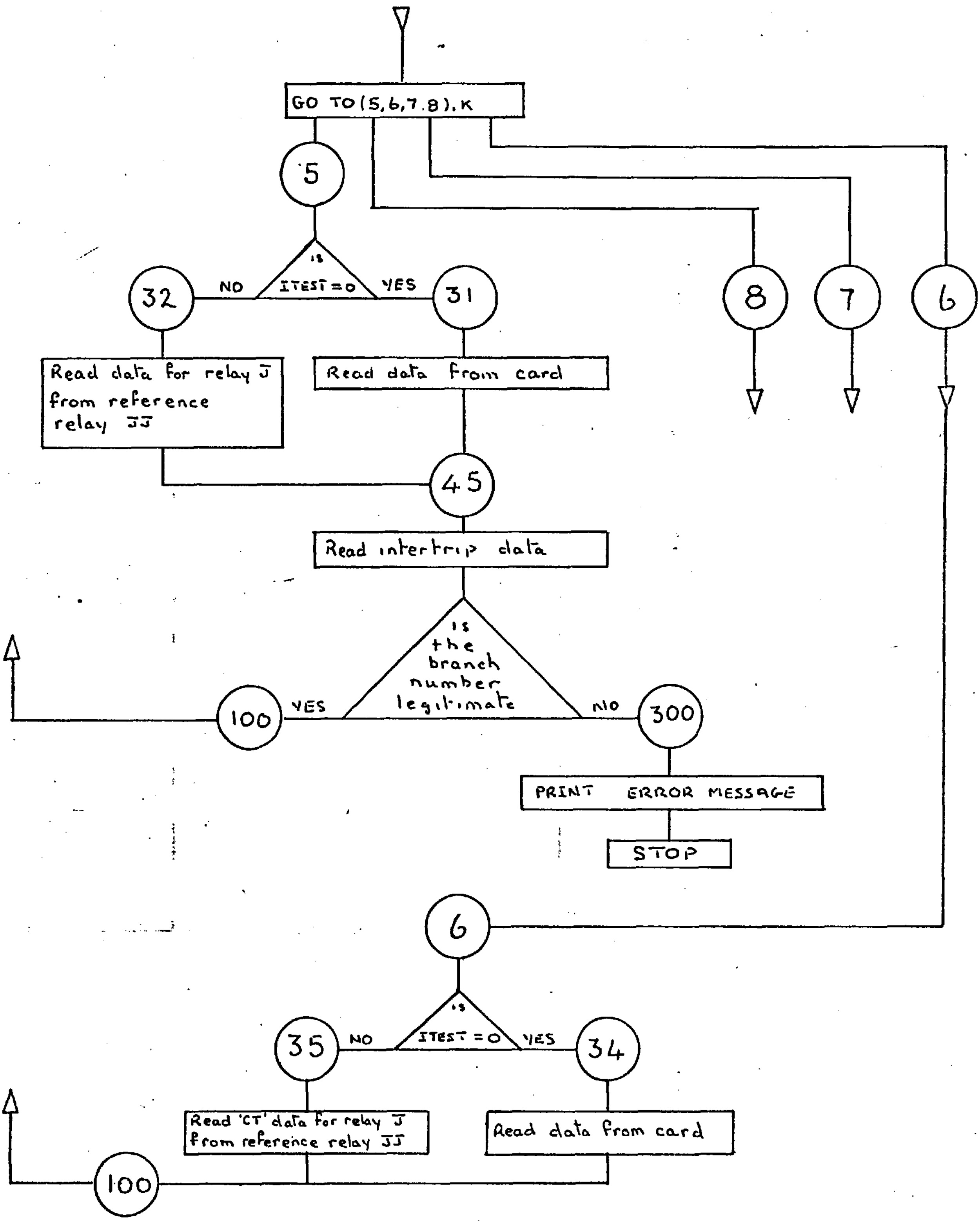
The relay data, even for a small system, is numerous and in order to facilitate the location of mispunched or incorrect data each relay data card has a data code punched in columns 3, 4, 5 and 6. These codes are fully described in chapter 5. The main function of RLDAT is to read the relay data and to distribute it to the working arrays and file storage.

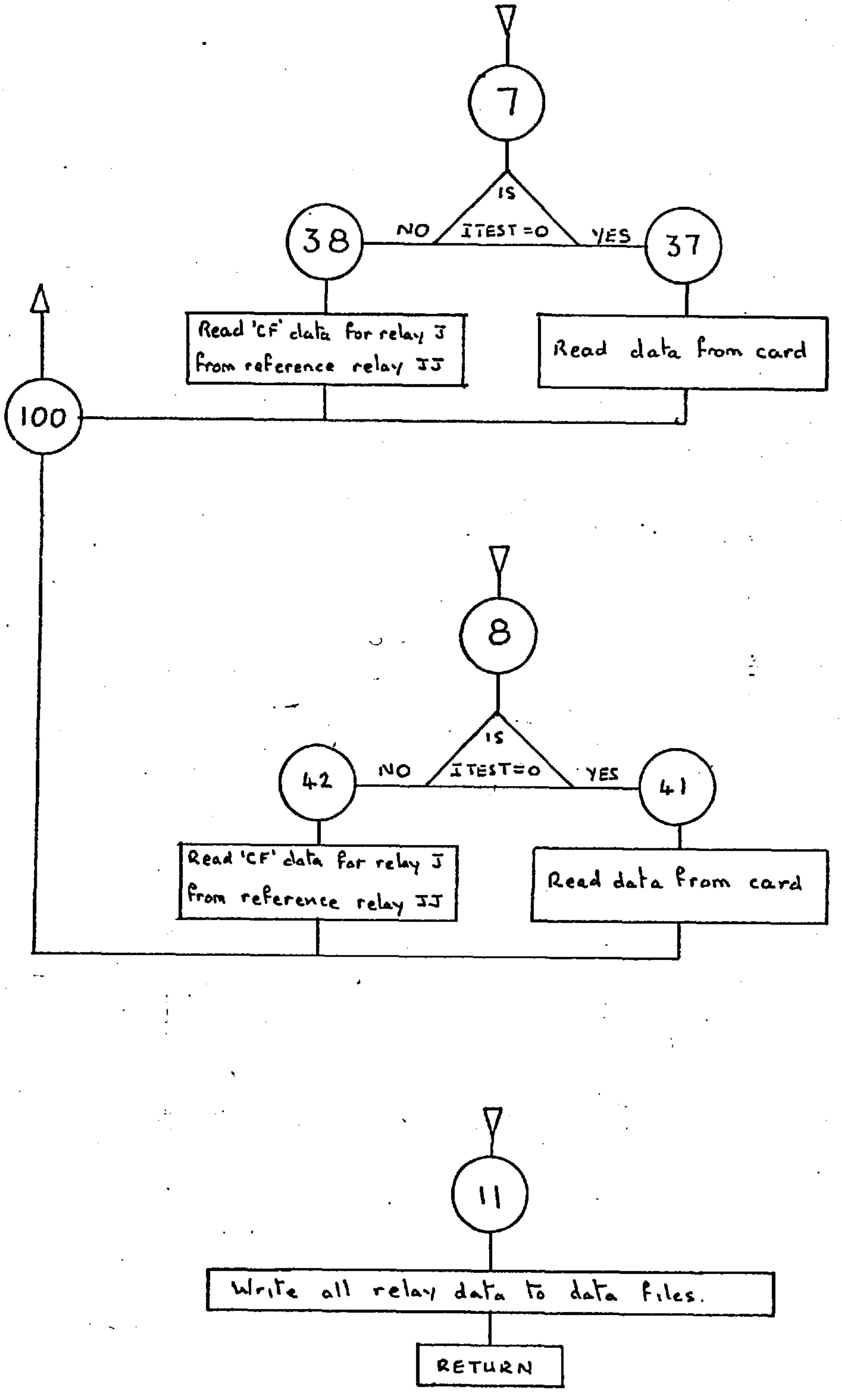
The machine oriented subroutine COMP is used by RLDAT to inspect the punched data codes and compare these, on a character basis, with the reference codes supplied by the Data Statements. The argument I, of COMP, returns the number of characters that are not identical in any given comparison. When a card is checked against all data codes and I is not returned as zero, the data card and a message 'Data code error' are printed on the line printer.

Subroutine RLDAT - Flow Chart




GO TO (16,17,18,11,16,17,18), K
11
14
PRINT the data card and message 'DATA CODE ERROR'
16
17
18
L=1
L=2
L=3
19
Read the relay number J
IS ITEST = 0
YES
NO
28
Read reference relay number JJ
IS JJ< 1 OR > NR
NO
YES
27
DO 3 K=1,4
Compare columns 3,4,5 & 6 of the data card with ZZZ(K) i.e. determine the type of data
IS I = 4 (Data identified)
YES
NO
4
3
PRINT data card, and message 'DATA CODE ERROR'
100


GO TO(5,6,7,8),K
5
IS ITEST = 0
NO
YES
32
31
8
7
6
Read data for relay J from reference relay JJ
Read data from card
45
Read intertrip data
is the branch number legitimate
YES
NO
100
300
PRINT ERROR MESSAGE
STOP
6
IS ITEST = 0
NO
YES
35
34
Read 'CT' data for relay J from reference relay JJ
Read data from card
100


7
IS
ITEST=0
NO
YES
38
37
Read 'CF' data for relay J
from reference relay JJ
Read data from card
100
8
IS
ITEST=0
NO
YES
42
41
Read 'CF' data for relay J
from reference relay JJ
Read data from card
11
Write all relay data to data files.
RETURN

6.18 Subroutine RLMOV

Computation of the relay movements commences at t = 0.01 seconds, which is the shortest time after the initiation of a fault that a reasonable estimate of the R.M.S. value of the fault current can be made, see 2.13.

In order to avoid the computation of a zero relay movement, two auxiliary arrays are used, TRIPSAVE and NOMOVE. TRIPSAVE records the state of the relays at the end of the nth interval ($n \neq 1$), and is used with relay TRIP, which records the state of the ~~relays~~ arrays at the end of the (n + 1)th interval. At the end of the (n + 1)th interval the corresponding elements of TRIP and TRIPSAVE are compared and if the state of a relay has remained unchanged during the period the appropriate element of NOMOVE is set to zero. When a relay moves during the period the corresponding element of NOMOVE is set equal to 1. Function MOVE see 6.38, is used to interrogate NOMOVE and determine the required course of action.

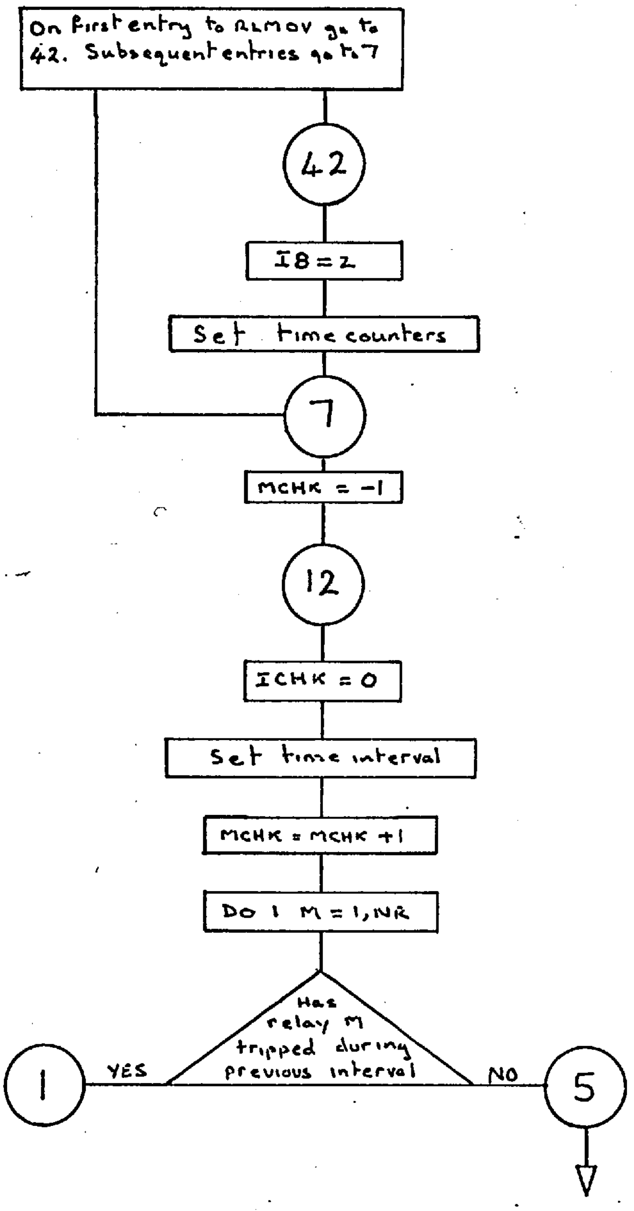
There are two main counters used in this routine, ICHK and MCHK. ICHK is normally zero, but is set to 1 when a relay trip has been recorded. This allows an exit to be made from RLMOV so that subroutine BREAK, see 6.31, can effect the required system changes and evaluate the new decremental equation coefficients, after which a return to RLMOV is made. MCHK is used, through common block C7, to indicate to function MOVE when the computations taking place relate to the first time interval. This is necessary, in order to avoid the interrogation of array

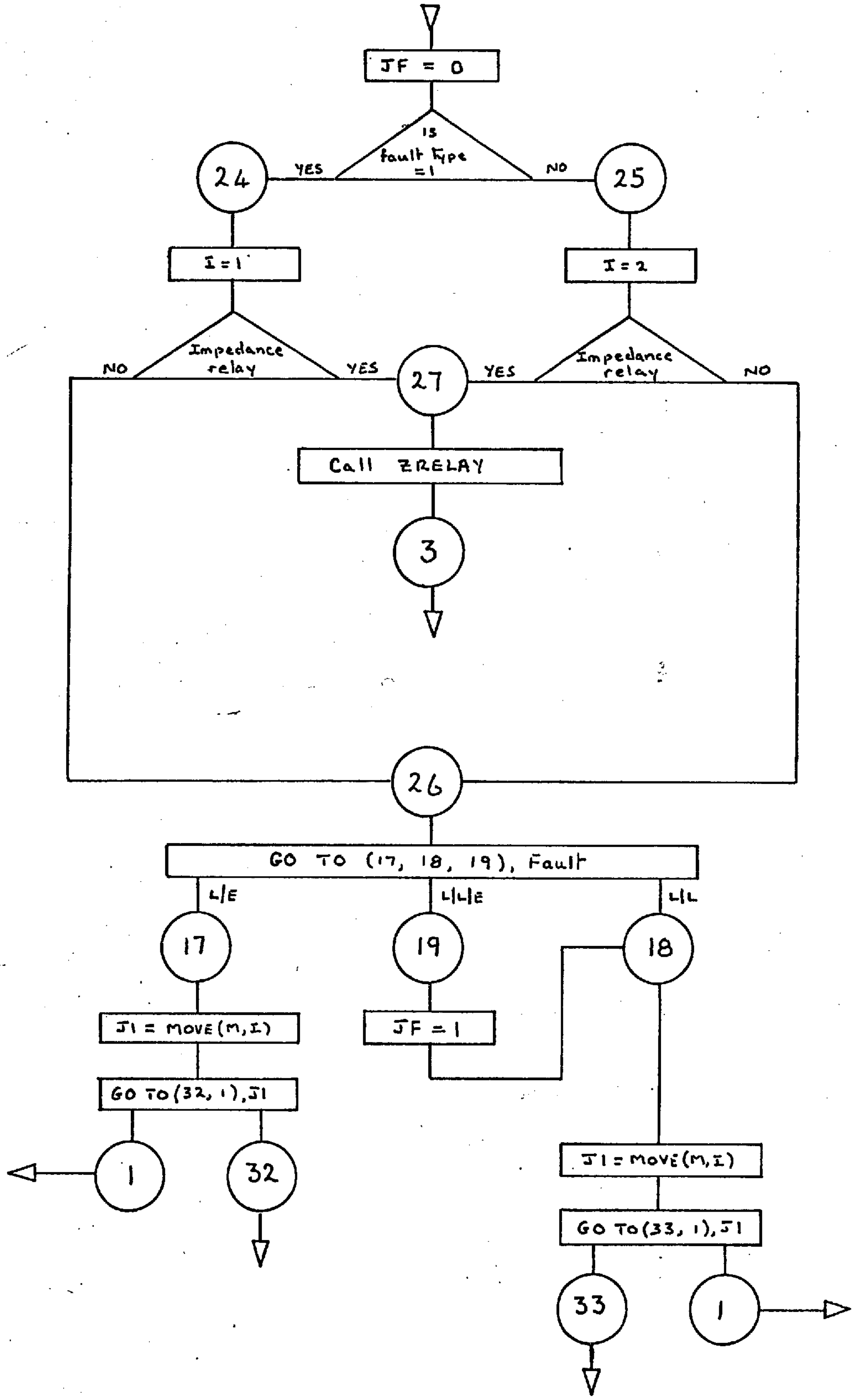
NOMOVE during the first time interval.

Array KOUNT records relay trips and has element values 0 for no trip, and 1 when a trip has taken place. This record allows the action required for tripped relays to be taken only once for any given relay.

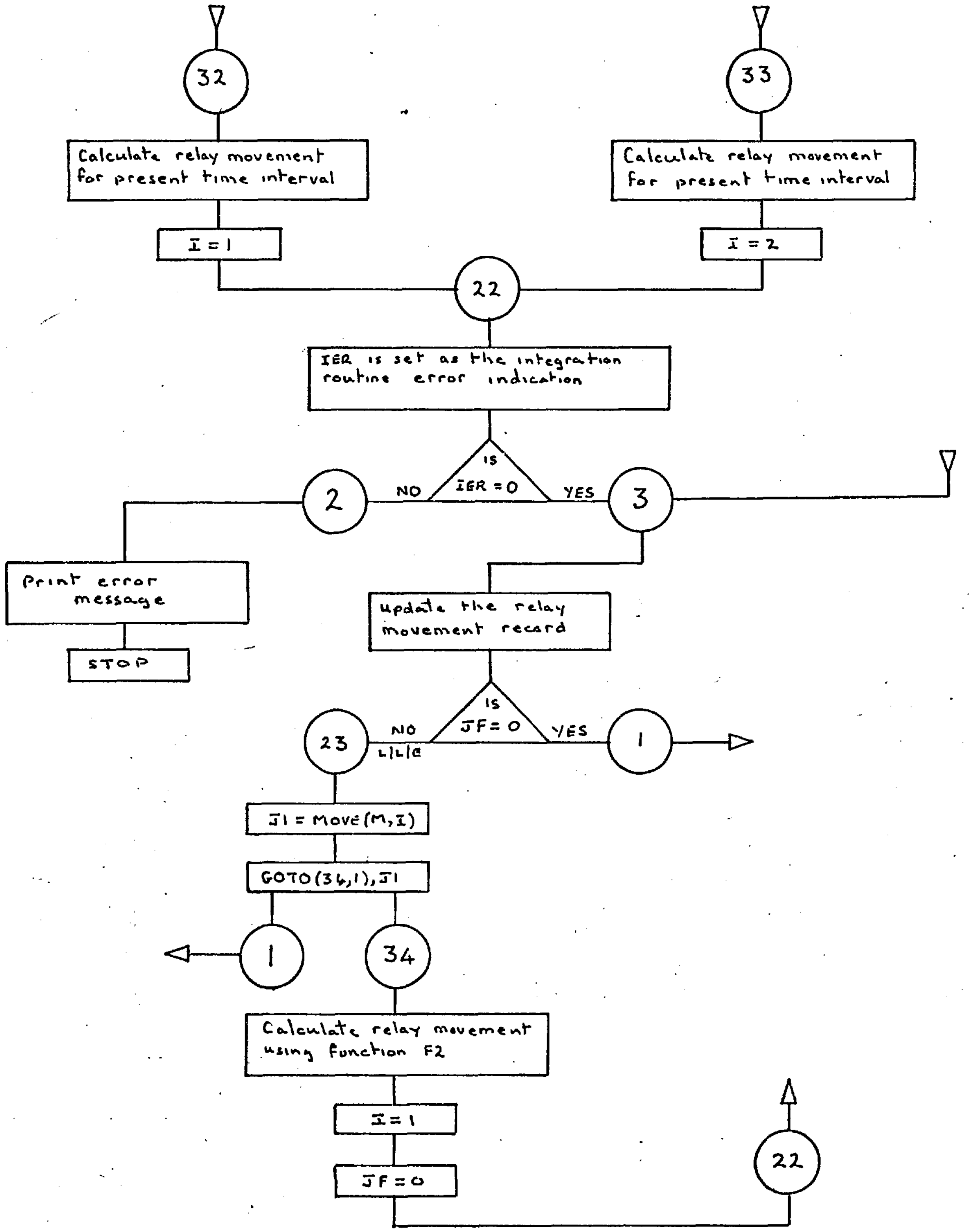
Function SMPSN is used by RLMOV to perform the integration, using Simpsons Rule, required to calculate the relay movement during the given time interval.

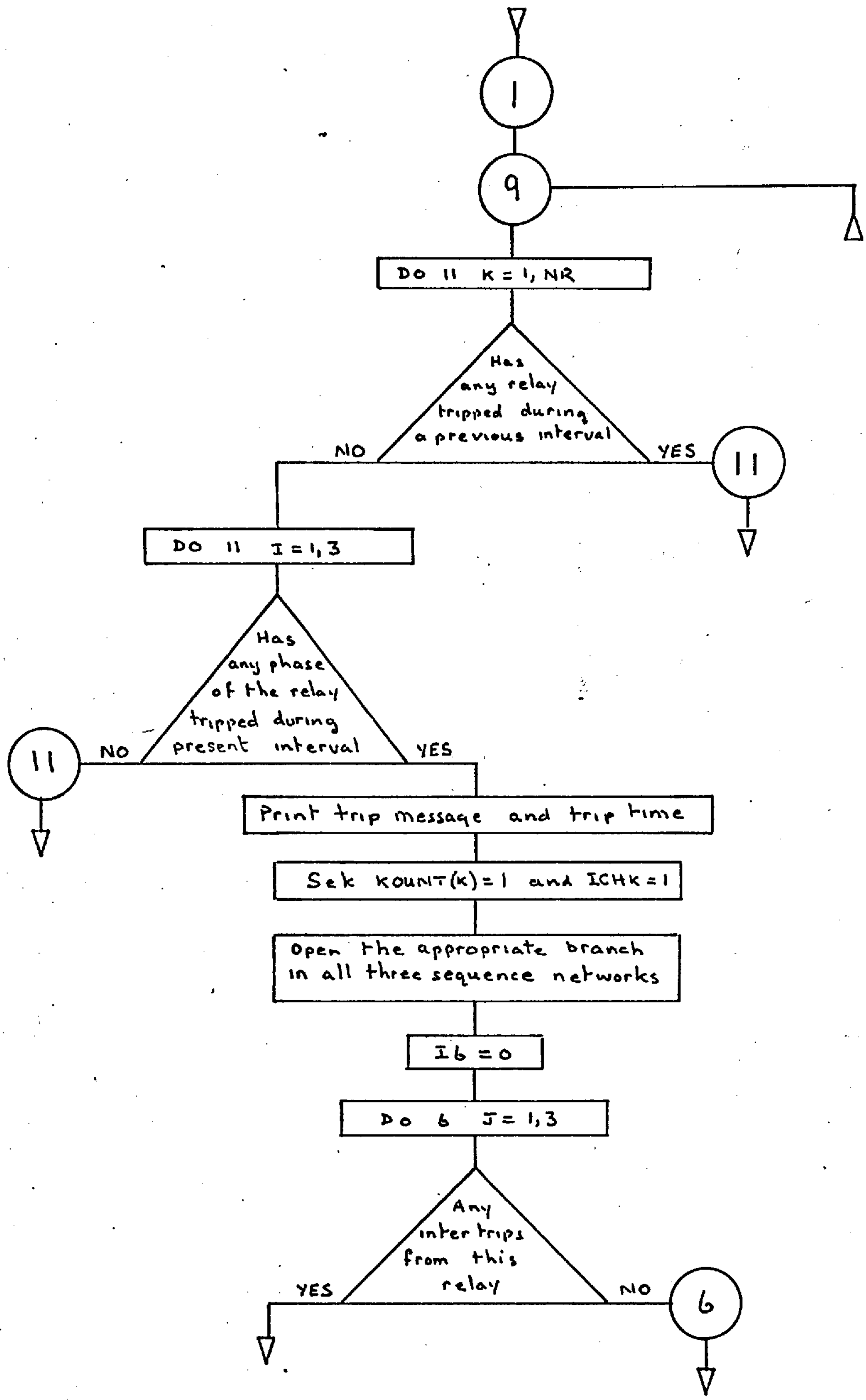
Subroutine RLMOV - Flow Chart

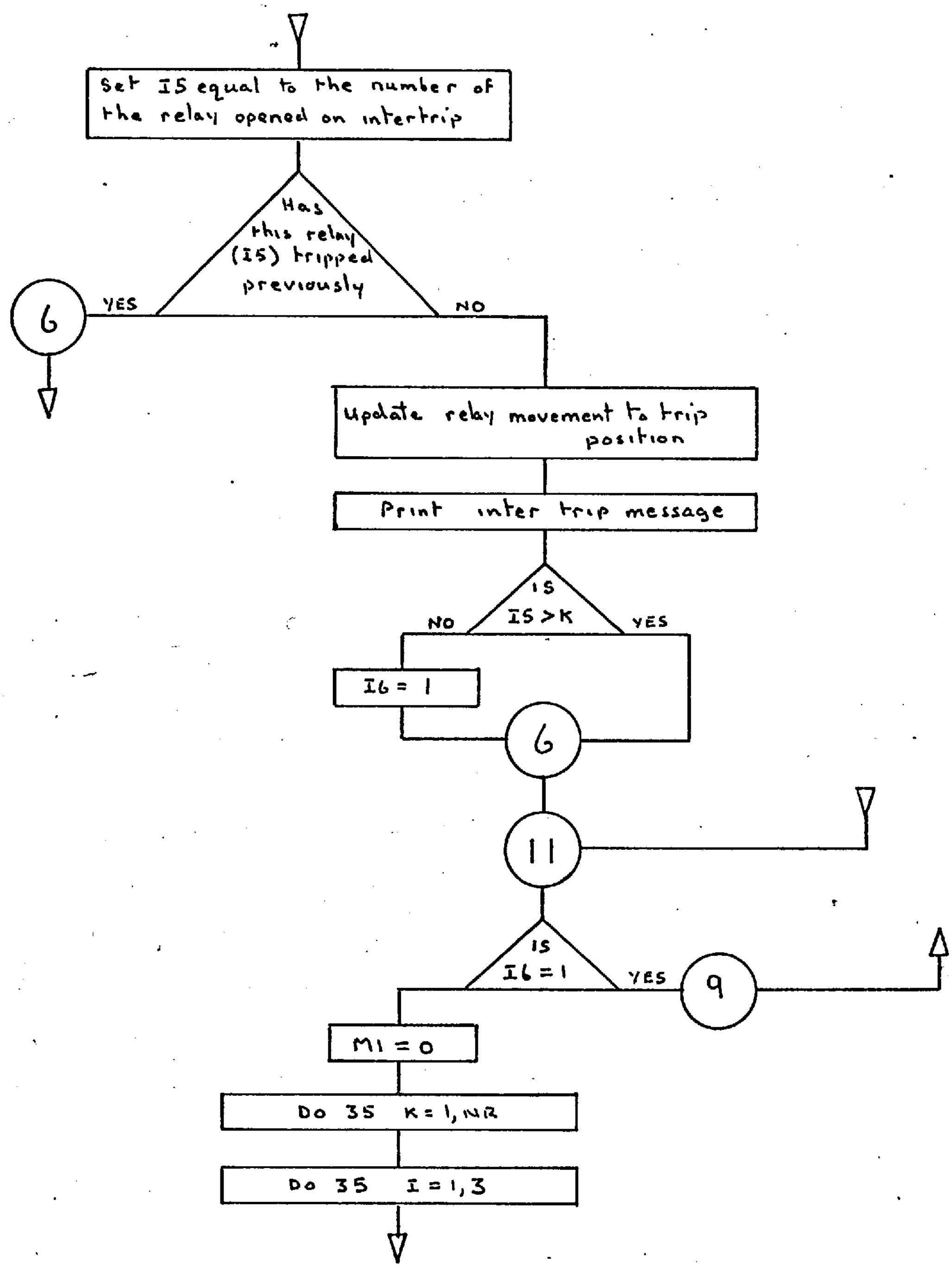


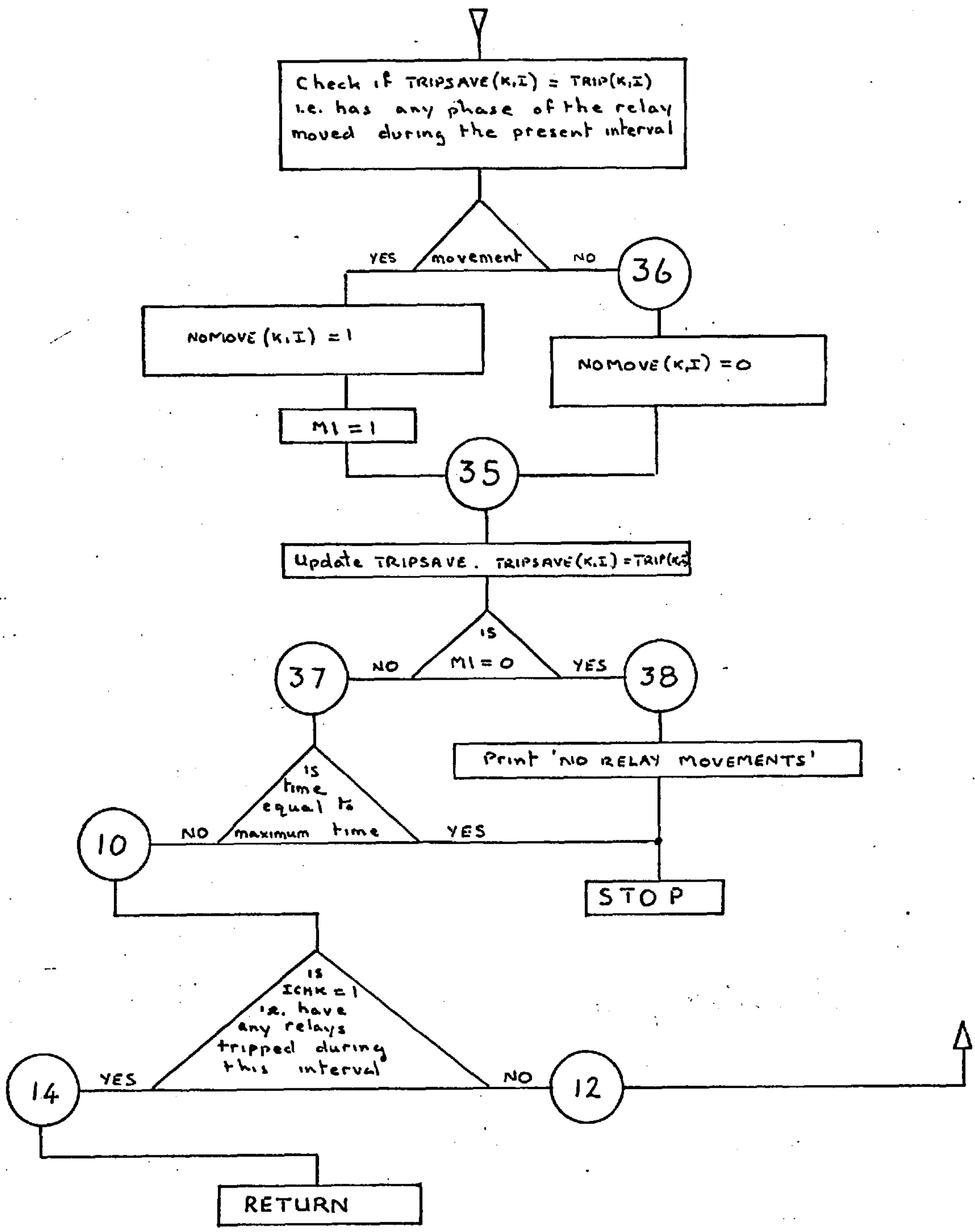


JF = 0
Is fault type = 1
YES
NO
24
25
I = 1
I = 2
Impedance relay
Impedance relay
NO
YES
YES
NO
27
Call ZRELAY
3
26
GO TO (17, 18, 19), Fault
L/E
L/L/E
L/L
17
19
18
J1 = MOVE(M,I)
JF = 1
GO TO(32, 1),J1
1
32
J1 = MOVE(M,I)
GO TO(33, 1),J1
33
1


32
Calculate relay movement for present time interval
I = 1
33
Calculate relay movement for present time interval
I = 2
22
IER is set as the integration routine error indication
IS IER = 0
NO
YES
2
3
Print error message
STOP
Update the relay movement record
IS JF = 0
NO
YES
23
1
J1 = Move(M,I)
GOTO(34,1),J1
1
34
Calculate relay movement using function F2
I = 1
JF = 0
22

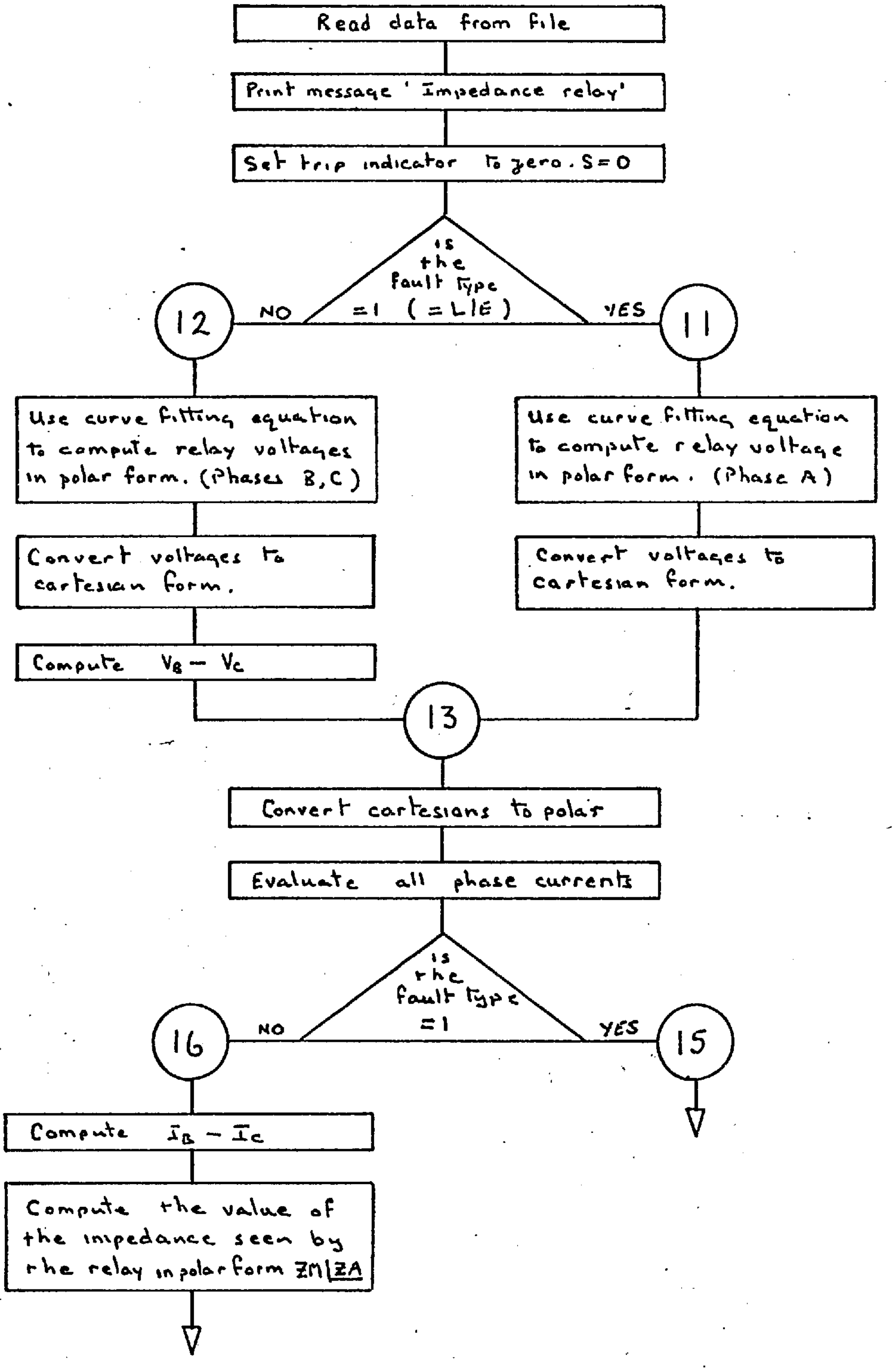

1
9
DO 11 K = 1, NR
Has any relay tripped during a previous interval
NO
YES
11
DO 11 I = 1,3
Has any phase of the relay tripped during present interval
11
NO
YES
Print trip message and trip time
Set KOUNT(K) = 1 and ICHK = 1
Open the appropriate branch in all three sequence networks
I6 = 0
Do 6 J = 1,3
Any inter trips from this relay
YES
NO
6

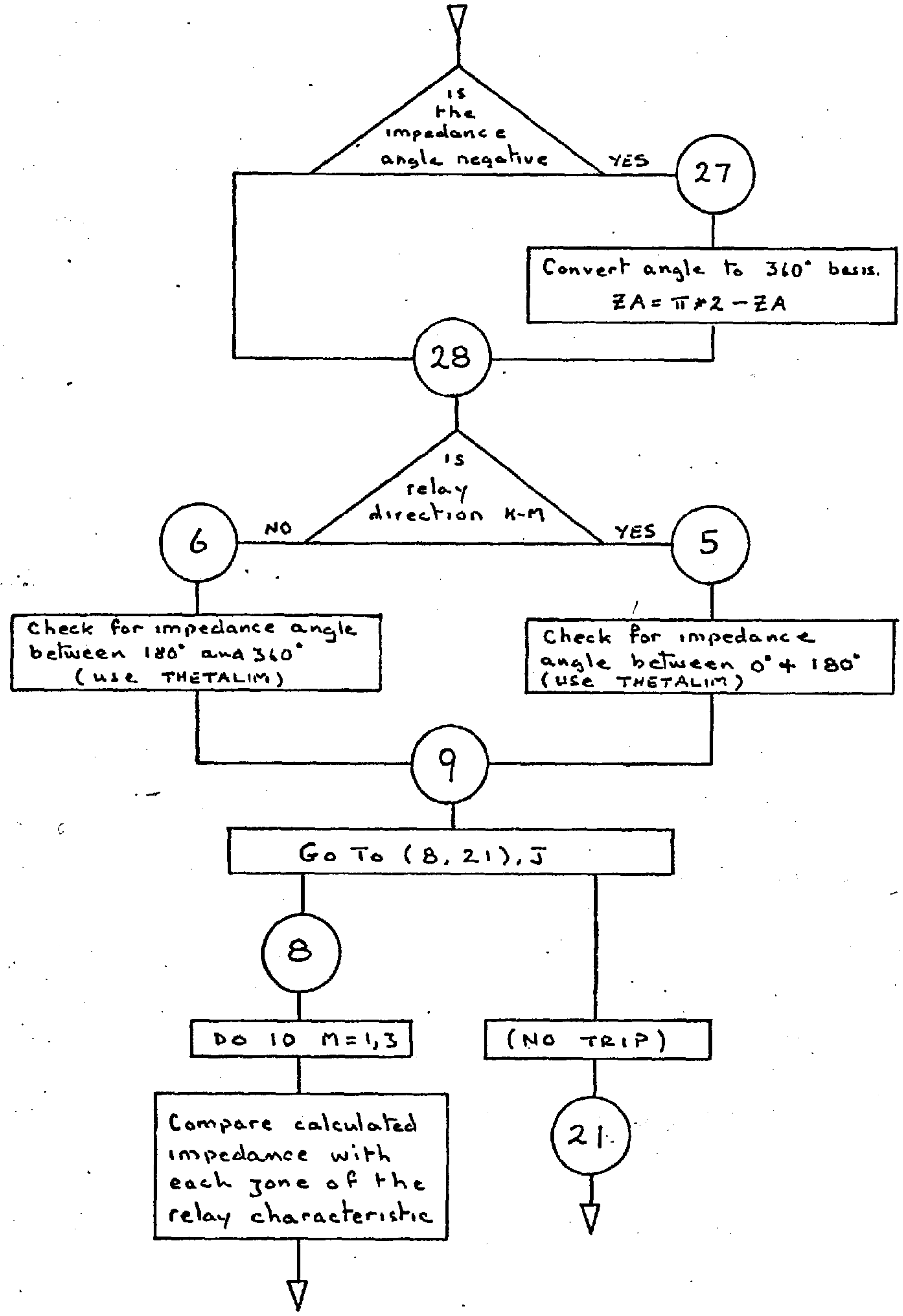

Set IS equal to the number of the relay opened on intertrip
Has this relay (IS) tripped previously
YES
NO
6
Update relay movement to trip position
Print inter trip message
IS IS > K
NO
YES
I6 = 1
6
11
IS I6 = 1
YES
9
M1 = 0
Do 35 K = 1, NR
Do 35 I = 1, 3

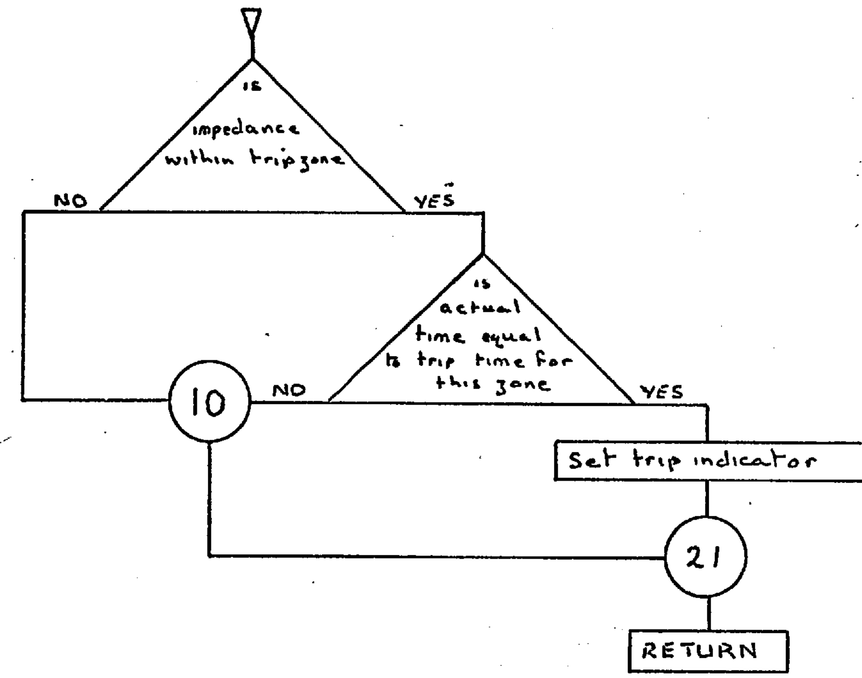

Check if TRIPSAVE(K,I) = TRIP(K,I) i.e. has any phase of the relay moved during the present interval
YES
movement
NO
36
NOMOVE(K,I) = 1
NOMOVE(K,I) = 0
M1 = 1
35
Update TRIPSAVE. TRIPSAVE(K,I) = TRIP(K,I)
NO
is M1 = 0
YES
37
38
Print 'NO RELAY MOVEMENTS'
10
NO
is time equal to maximum time
YES
STOP
is ICHK = 1 i.e. have any relays tripped during this interval
14
YES
NO
12
RETURN

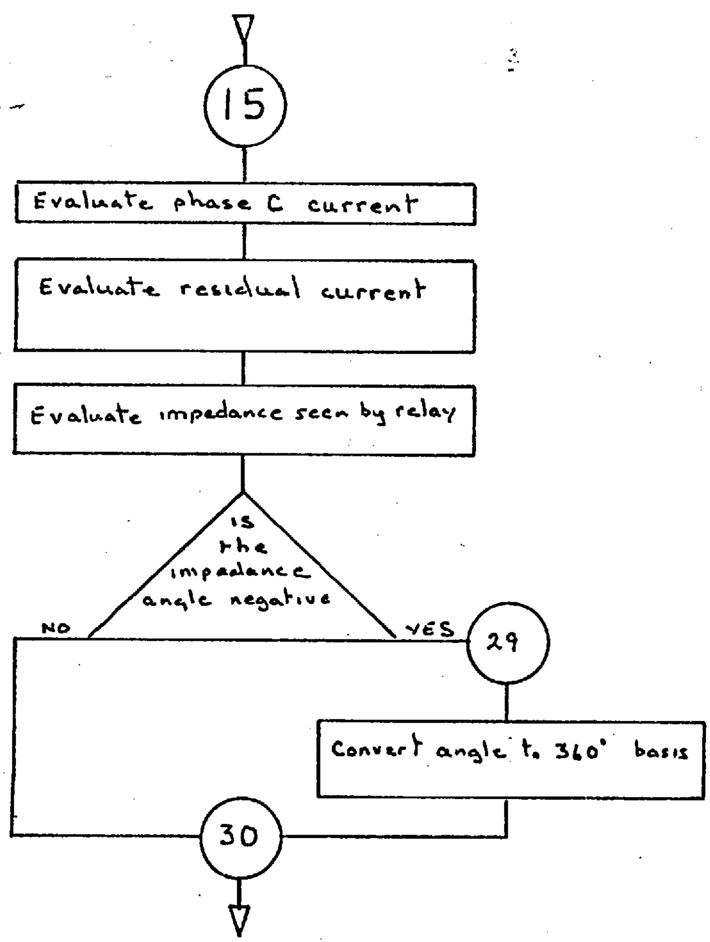
6.19

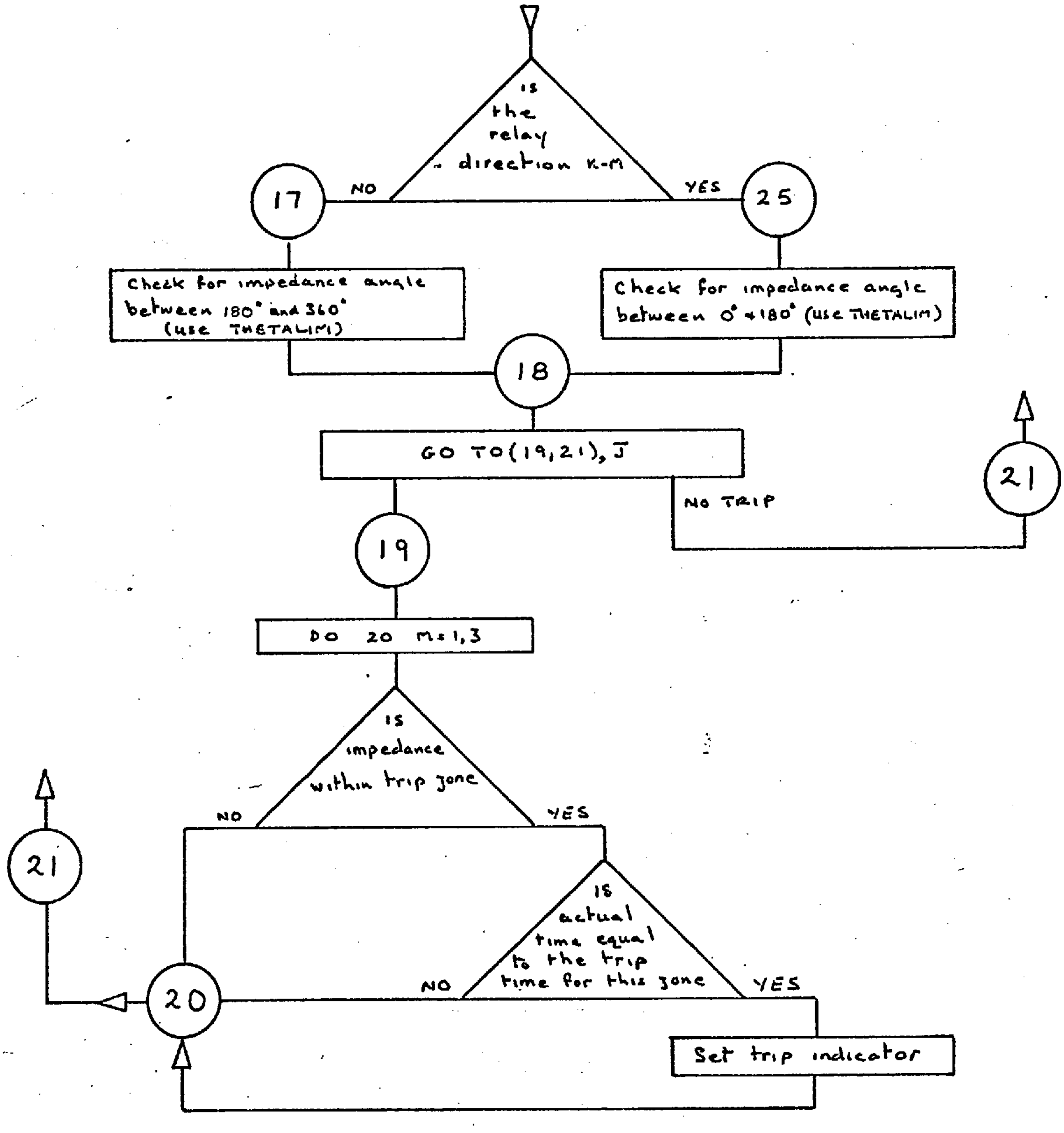
Subroutine ZRELAY - Flow Chart




is the impedance angle negative
YES
27
Convert angle to 360° basis.
ZA = π*2 − ZA
28
is relay direction K-M
NO
YES
6
5
Check for impedance angle between 180° and 360° (use THETALIM)
Check for impedance angle between 0° + 180° (use THETALIM)
9
Go To (8, 21), J
8
DO 10 M=1,3
(NO TRIP)
Compare calculated impedance with each zone of the relay characteristic
21


is impedance within trip zone
NO
YES
is actual time equal to trip time for this zone
10
NO
YES
Set trip indicator
21
RETURN


15
Evaluate phase C current
Evaluate residual current
Evaluate impedance seen by relay
is the impedance angle negative
NO
YES
29
Convert angle to 360° basis
30


is the relay direction K-M
NO
YES
17
25
Check for impedance angle between 180° and 360° (use THETALIM)
Check for impedance angle between 0° + 180° (use THETALIM)
18
GO TO(19,21), J
NO TRIP
21
19
DO 20 M=1,3
is impedance within trip zone
NO
YES
21
is actual time equal to the trip time for this zone
NO
YES
20
Set trip indicator

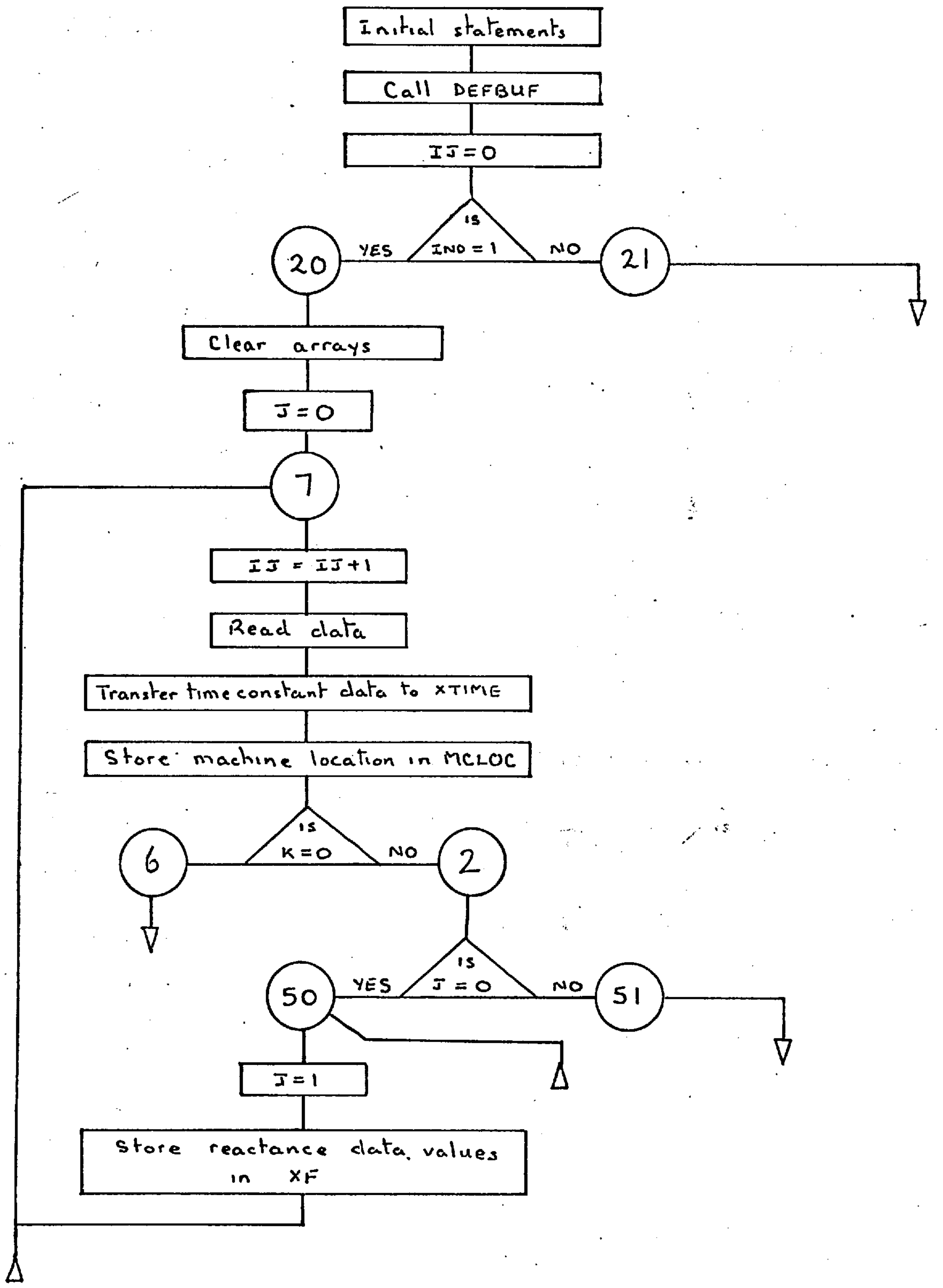
6.20 Subroutine MODS(IND)

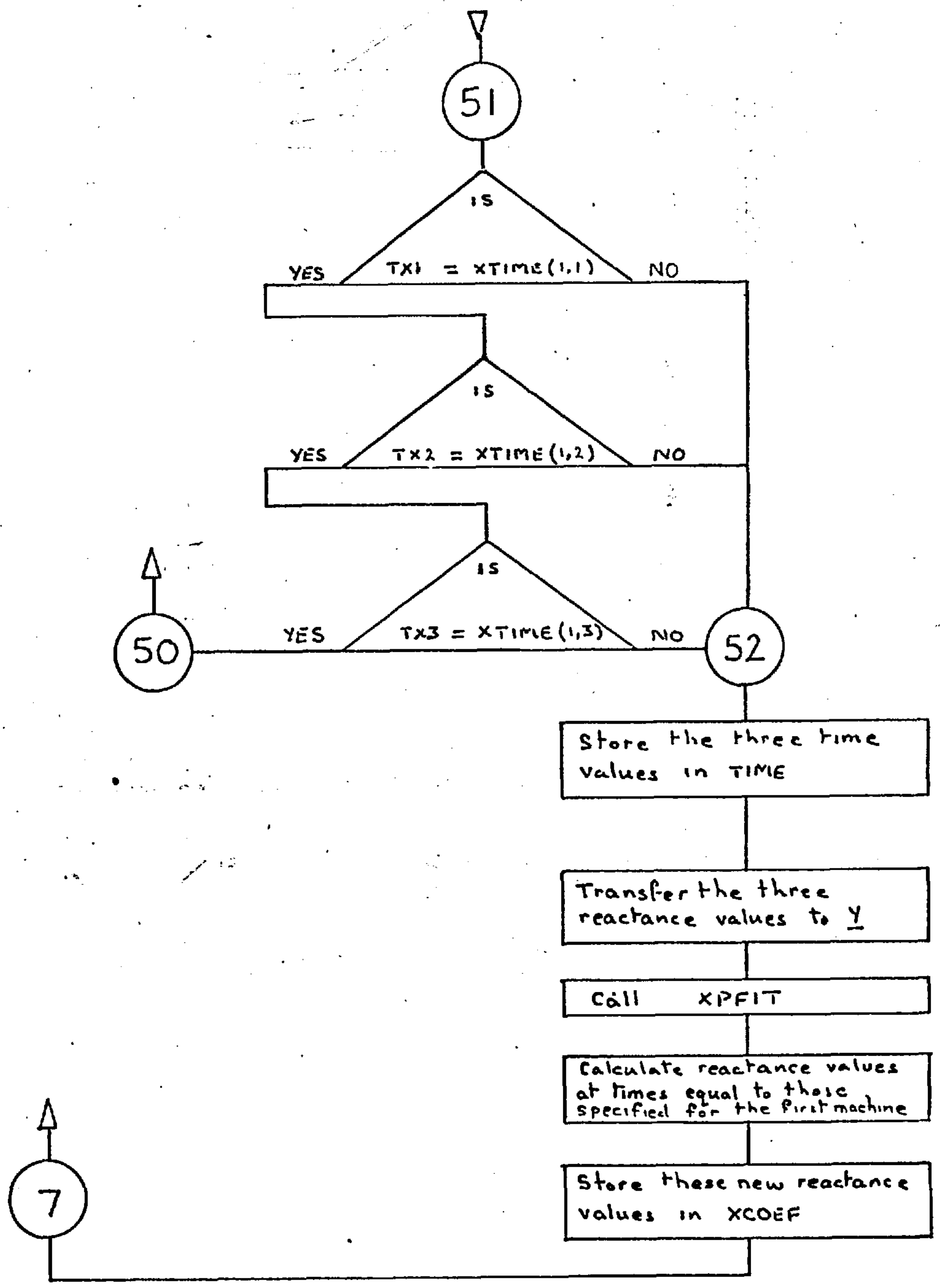
Subroutine MODS is used to read the machine reactance data, including the time constants, and also to modify the P.S.N. so as to include appropriate values of machine reactance. MODS is called three times, and the argument variable IND indicates the number of the call, which dictates the action to be taken.

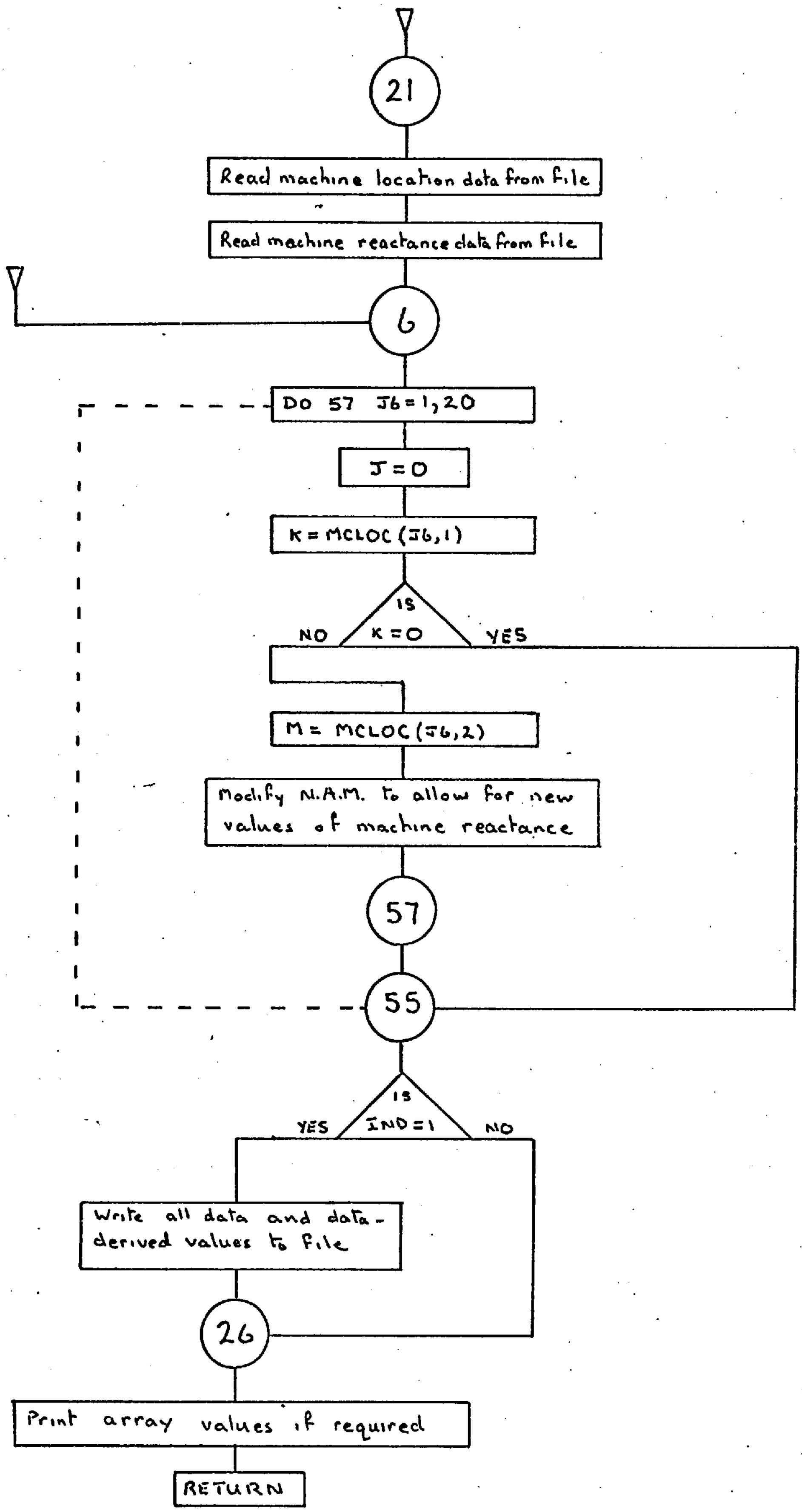
As can be seen from the flow chart, this is an involved procedure since all machines are unlikely to have the same time constants and/or values of reactance. The data carried by the first data card in this section is taken as reference data. Hence this data should relate to the machine with the shortest time constants. Consider a system containing two machines A and B, if the first card contains the data for machine A this data is stored directly in array XF. When machine B has the same time constants as machine A, the data for B is also stored directly in XF. Three analyses are then performed at network times T_1, T_2 and T_3. The results of these analyses are then used by the curve fitting routines to determine equations for the system voltages and currents.

When the time constants of machine B are different to those of machine A, the above procedure is not immediately possible. Values of reactance for machine B have to be computed at times corresponding to the time constants of machine A. This is achieved by evaluating the coefficients of the reactance equation for machine B, and calculating the reactances at the required times. The three system analyses are then performed to provide data for the curve fitting routines.

Subroutine MODS - Flow Chart




51
IS
TX1 = XTIME(1,1)
YES
NO
IS
TX2 = XTIME(1,2)
YES
NO
IS
TX3 = XTIME(1,3)
YES
NO
50
52
Store the three time values in TIME
Transfer the three reactance values to Y
Call XPFIT
Calculate reactance values at times equal to those specified for the first machine
Store these new reactance values in XCOEF
7


21
Read machine location data from file
Read machine reactance data from file
6
DO 57 J6=1,20
J=0
K=MCLOC(J6,1)
IS K=0
NO
YES
M= MCLOC(J6,2)
Modify N.A.M. to allow for new values of machine reactance
57
55
IS IND=1
YES
NO
Write all data and data-derived values to file
26
Print array values if required
RETURN

6.21 Subroutine OPEN(K,N1,N2)

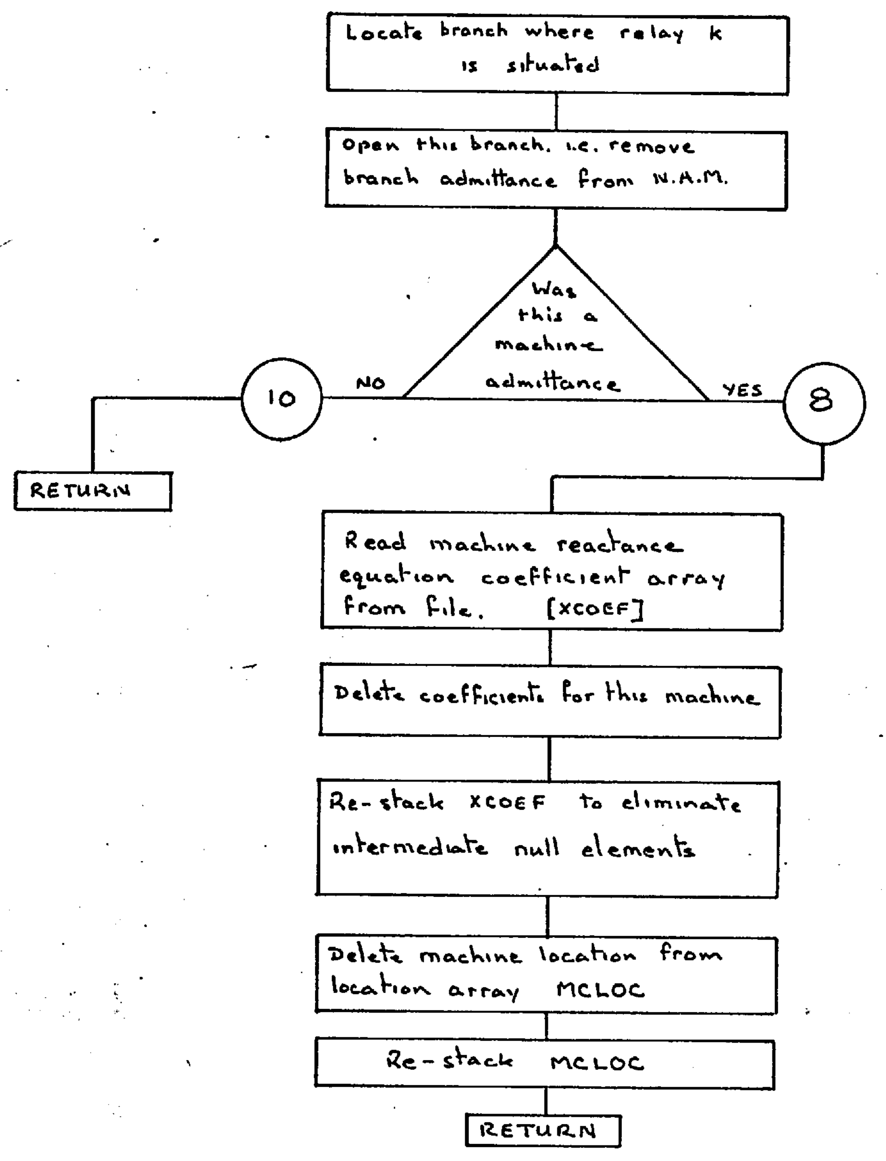
K is the relay position

N1 is the file number for the conductance array

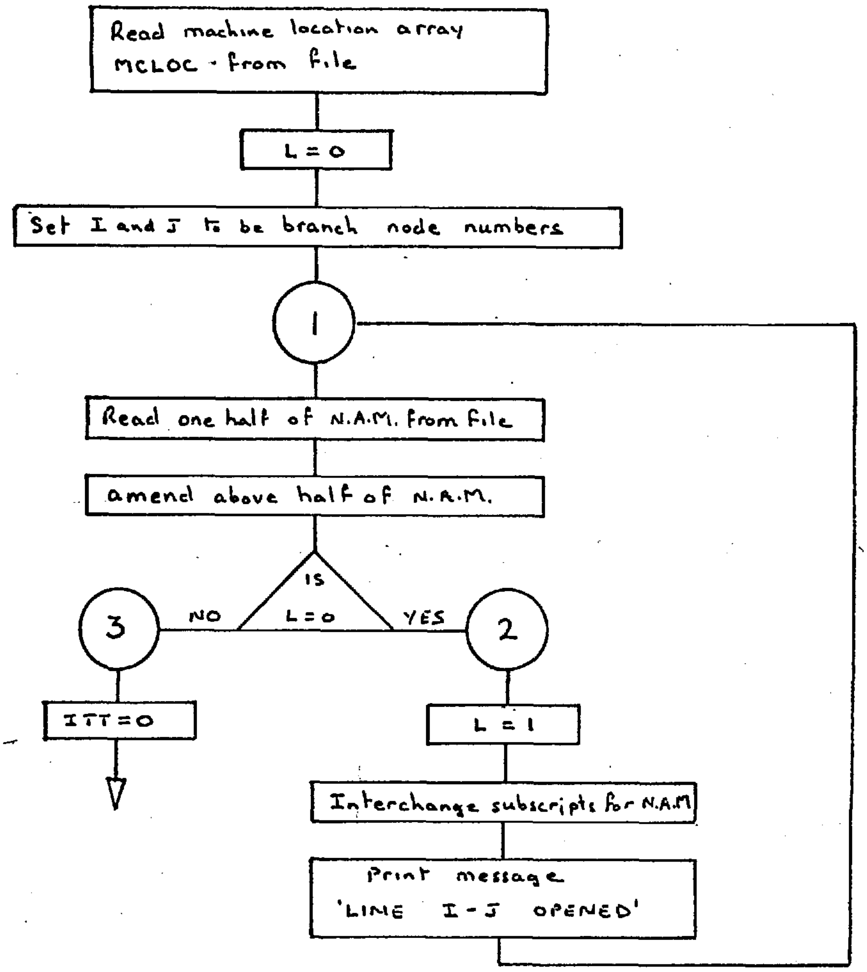
N2 is the file number for the susceptance array

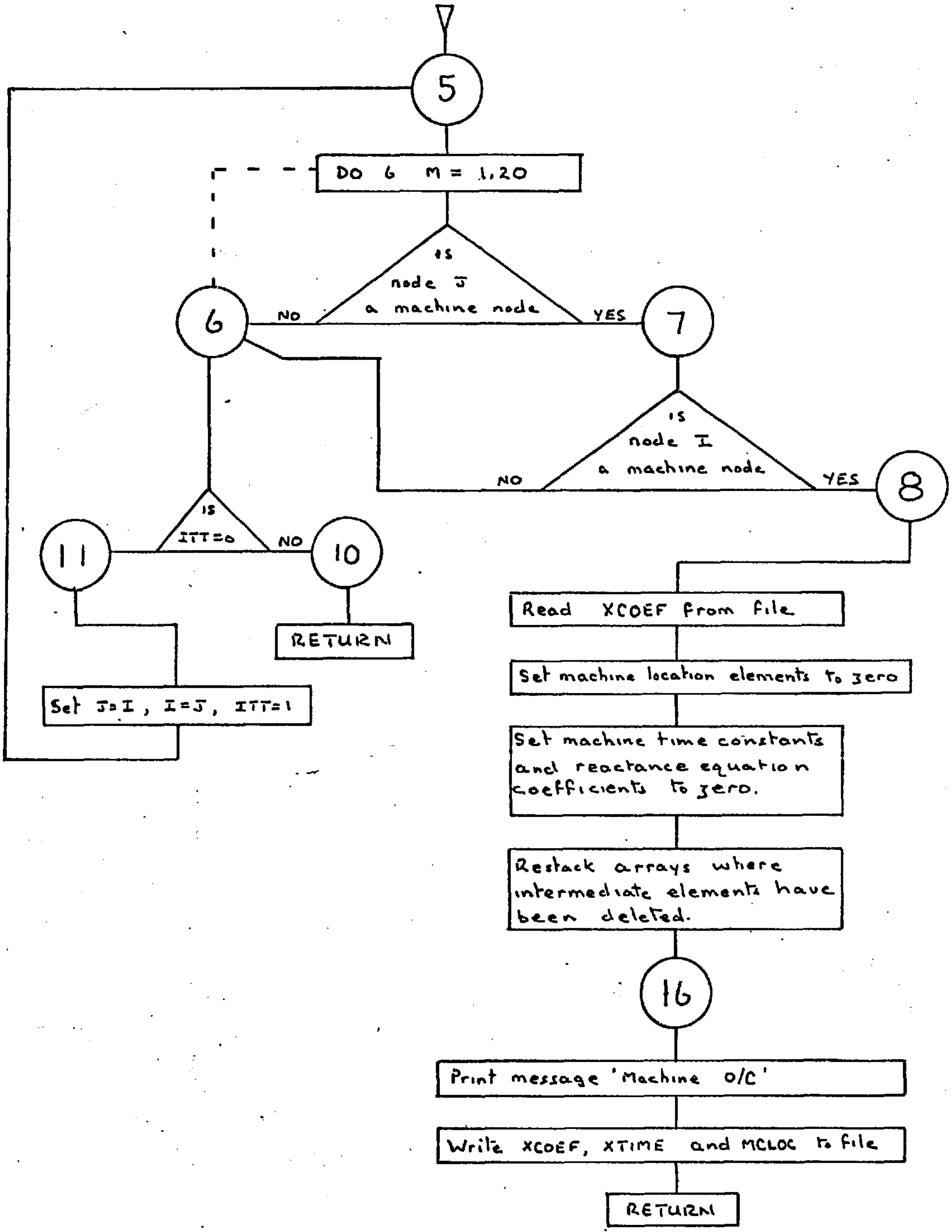
In order to simulate circuit breaker action, lines (or the admittance values of lines) have to be removed from the admittance matrices. Since the admittance matrices are stored in two parts, G and B, OPEN is supplied with two arguments, N1 and N2, so that the required files can be accessed, see 6.47. For normal system lines the procedure is straightforward, Y(K,M) and Y(M,K) are set to zero and the value of Y(K,K) is recomputed. However, some of the elements of the nodal admittance matrix represent machine reactances which also have an associated set of arrays; MCLOC (machine location), XCOEF (machine reactance equation coefficients) and XTIME (machine time constants). It is necessary, therefore, to ascertain if an element which is to be deleted, is a machine reactance element so that the corresponding values of MCLOC, XCOEF and XTIME can also be set to zero. When an intermediate element of these arrays is set to zero the arrays are restacked so as to eliminate the intermediate null elements.

Subroutine OPEN - Macro Flow Chart



Subroutine OPEN - Flow Chart



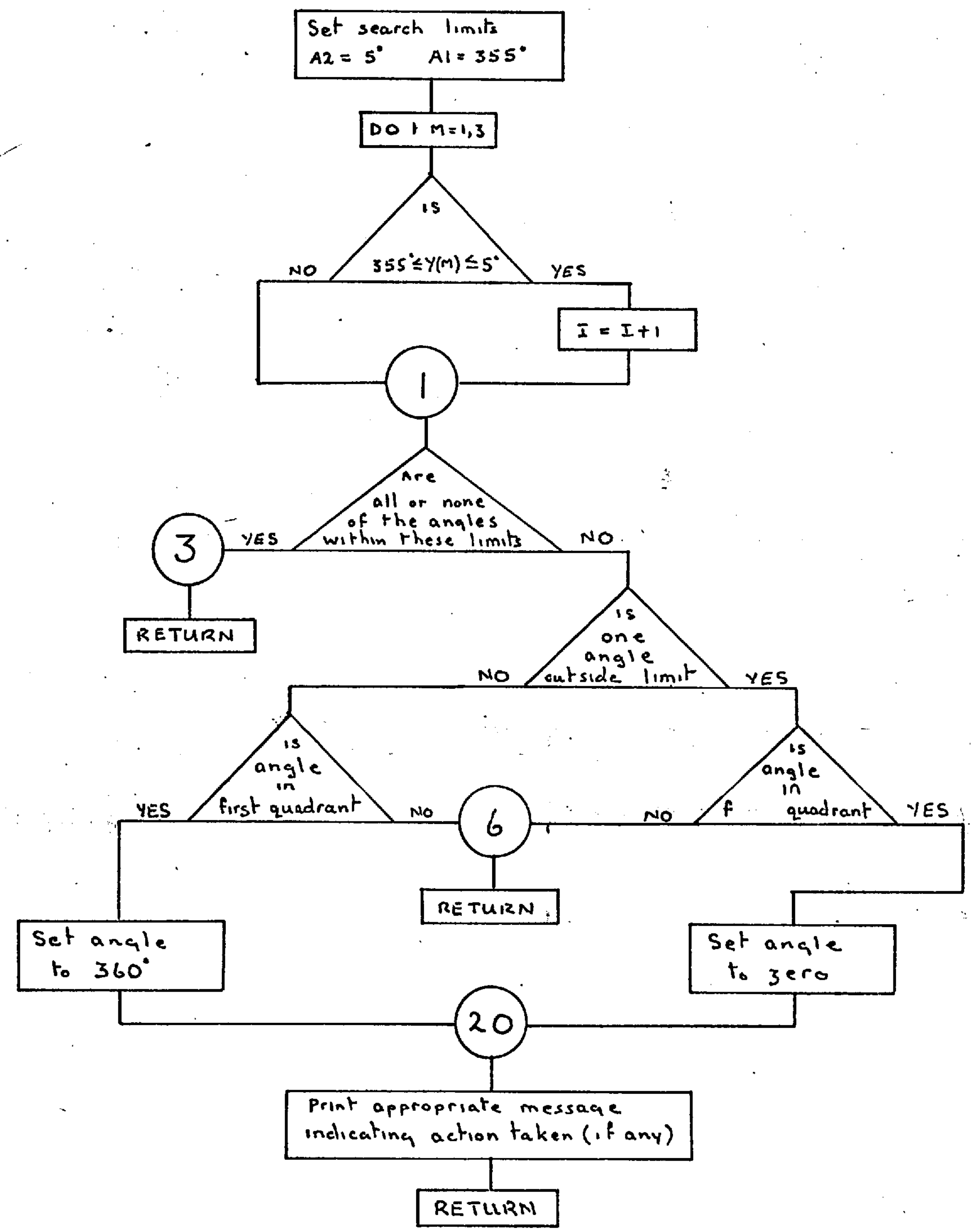

5
DO 6 M = 1,20
IS node J a machine node
NO
YES
6
7
IS node I a machine node
NO
YES
8
IS ITT=0
NO
11
10
RETURN
Set J=I, I=J, ITT=1
Read XCOEF from file
Set machine location elements to zero
Set machine time constants and reactance equation coefficients to zero.
Restack arrays where intermediate elements have been deleted.
16
Print message 'Machine O/C'
Write XCOEF, XTIME and MCLOC to file
RETURN

6.22 Subroutine CHECKANG(Y,K,I1)

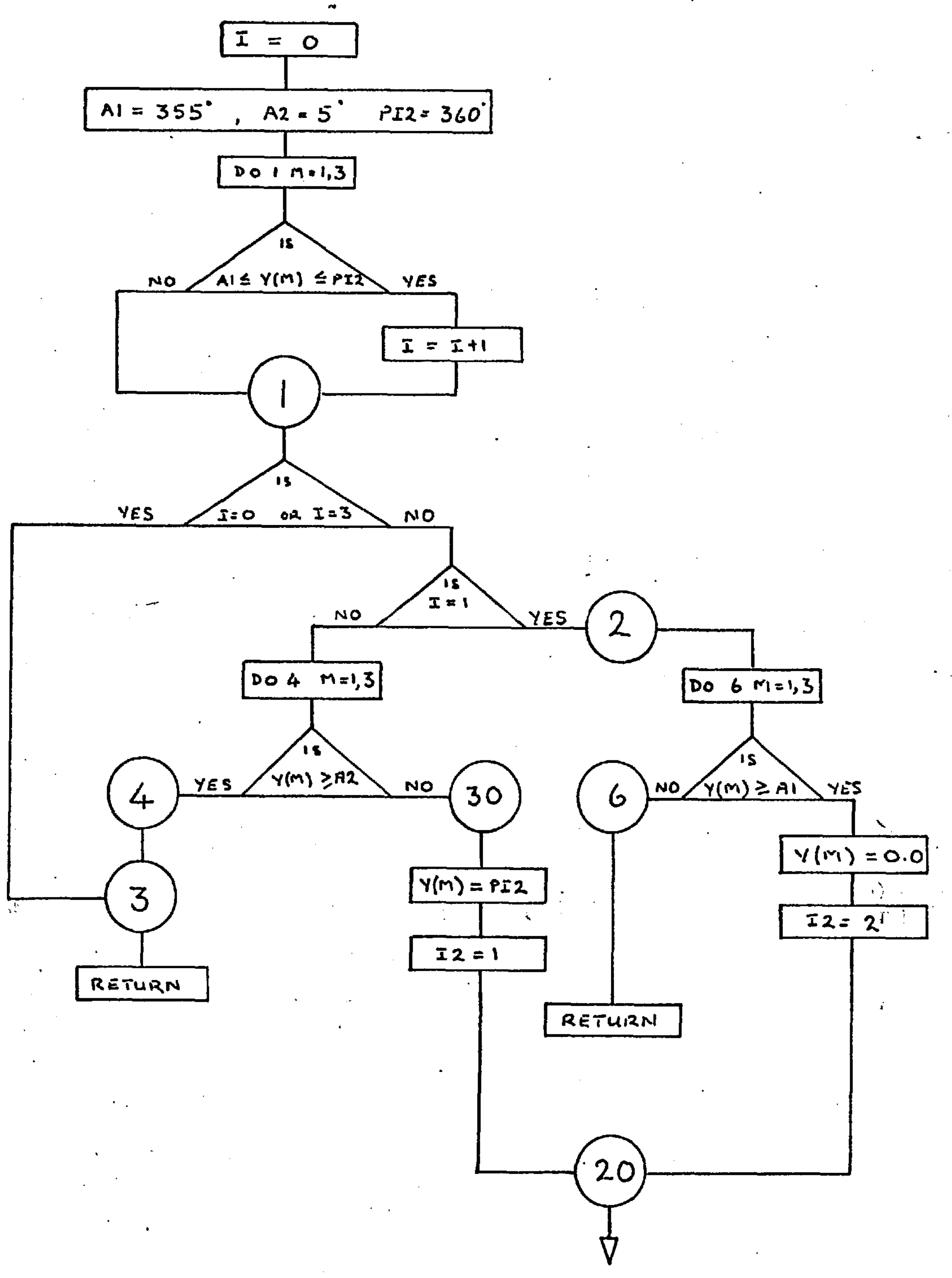
Before the current or voltage phase angle equations can be determined, the data has to be checked to see if it is compatible with the curve fitting routines. This check is necessary because experimental error can produce values which are inconsistent as far as the curve fitting routines are concerned. This can only happen when the angles are in the region of zero or 360 degrees. For example, the three data values could be 0.8, 359.6 and 0.8 degrees, CHECKANG will adjust the second data point to zero degrees. Should the situation be reversed, 359.6, 0.8 and 359.6 degrees, then CHECKANG will adjust the second data point to 360 degrees.

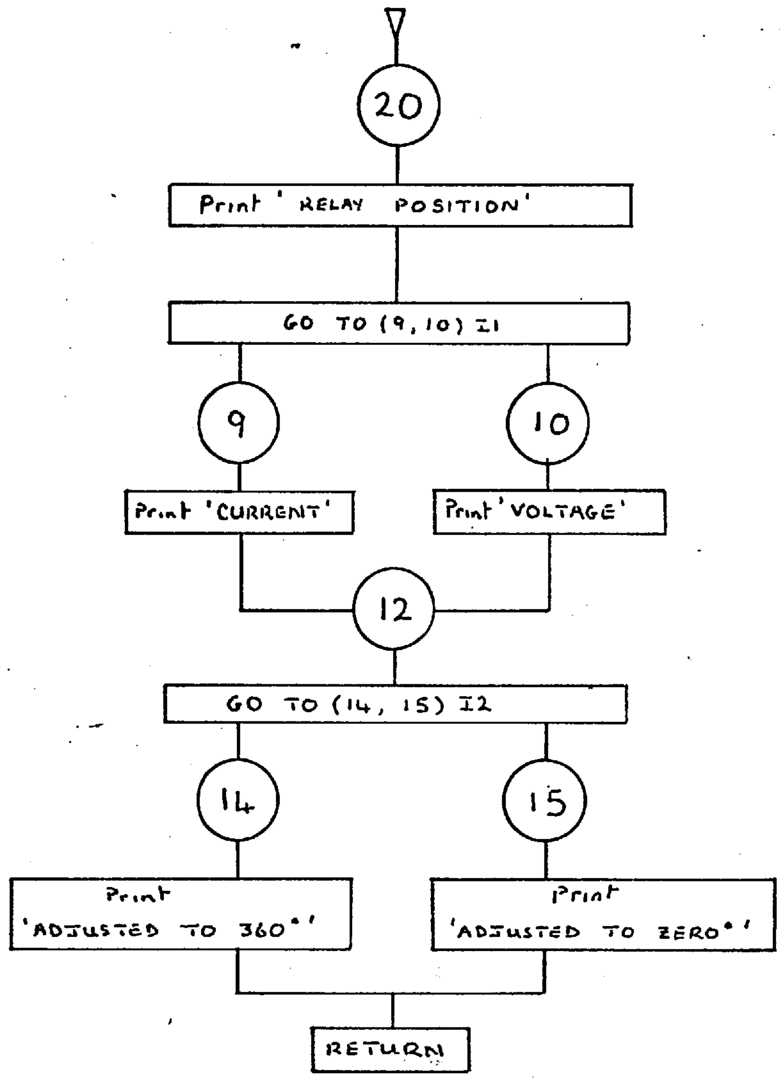
Experience has shown that the maximum difference is of the order of 1.5 degrees, however CHECKANG has an area of search of ± 5 degrees from zero. Should the spread of angles be outside this range, there is a data error, which will be signified by an appropriate message from thecurve fitting routines; XPEQU, XPFIT, VEQU, CALC.

Subroutine CHECKANGLE - Macro Flow Chart



<u>Subroutine CHECKANGLE - Flow Chart</u>




20
Print 'RELAY POSITION'
GO TO (9, 10) I1
9
10
Print 'CURRENT'
Print 'VOLTAGE'
12
GO TO (14, 15) I2
14
15
Print 'ADJUSTED TO 360°'
Print 'ADJUSTED TO ZERO°'
RETURN

6.23 Subroutine CREAD

This routine is used to read the data cards, in Format 10A8, and to transfer each as one record to the basic data file - file 22.

6.24 Subroutine FREAD

This routine is used to read the records of file 22, it therefore simulates the reading of the data cards. If a PRINT DATA card is included in the program control section, see 5.6, FREAD will print each record (card) as it is used by the program.

6.25 Subroutine ERROR(K)

Subroutine ERROR is used when a 'fatal' error is encountered. An error message is printed on the line printer and the program run is terminated. The error message is accompanied by an error number and a list of these numbers, with explanations, is given in 6.45.

6.26 <u>Subroutine COMP(I, A, J, B, K)</u>

This routine is used to compare two strings of characters for equality. In this program, one of the character strings is always supplied by a Data Statement in the subroutine which calls COMP and the other data string is obtained from the data card which is being identified.

I	is the number of characters to be compared.
A	contains one of the character strings.
J	is the number of the character in A from which comparison is to start.
B	contains the second character string.
K	is the number of the character in B from which comparison is to start.

On return from this routine I is set equal to the number of identical pairs of characters before which the first non-identical pair is encountered.

This routine is used by subroutines DATACNTRL and RLDAT.

6.27 Subroutine DEFBUF (N, I, A)

This subroutine is used to specify an array that will be associated with a specific channel number, so that READ or WRITE statements referring to that channel will cause transfers of records to and from the array instead of to and from a peripheral device.

Thus, FREAD is used to read a record from the basic data file into array A, array A is then treated as if it were a data card and data is then transferred from the array with a normal read format statement.

6.28 Subroutine DAM(I1, I2, I3)

This subroutine is used to print the first ten values held in the current data files I1, I2 and I3. This routine is a debugging aid which enables the above files to be interogated at various stages of the program. DAM is activated by the trace variable LEVEL, see 6.44.

6.29 Subroutine ACCL1 (IB, IX, I)

This subroutine is used to select and store five consecutive sets of nodal voltages which are used by the acceleration routine PADE.

IB is the number of the iteration at which selection is to commence.

IX counts the number of sets of voltages stored.

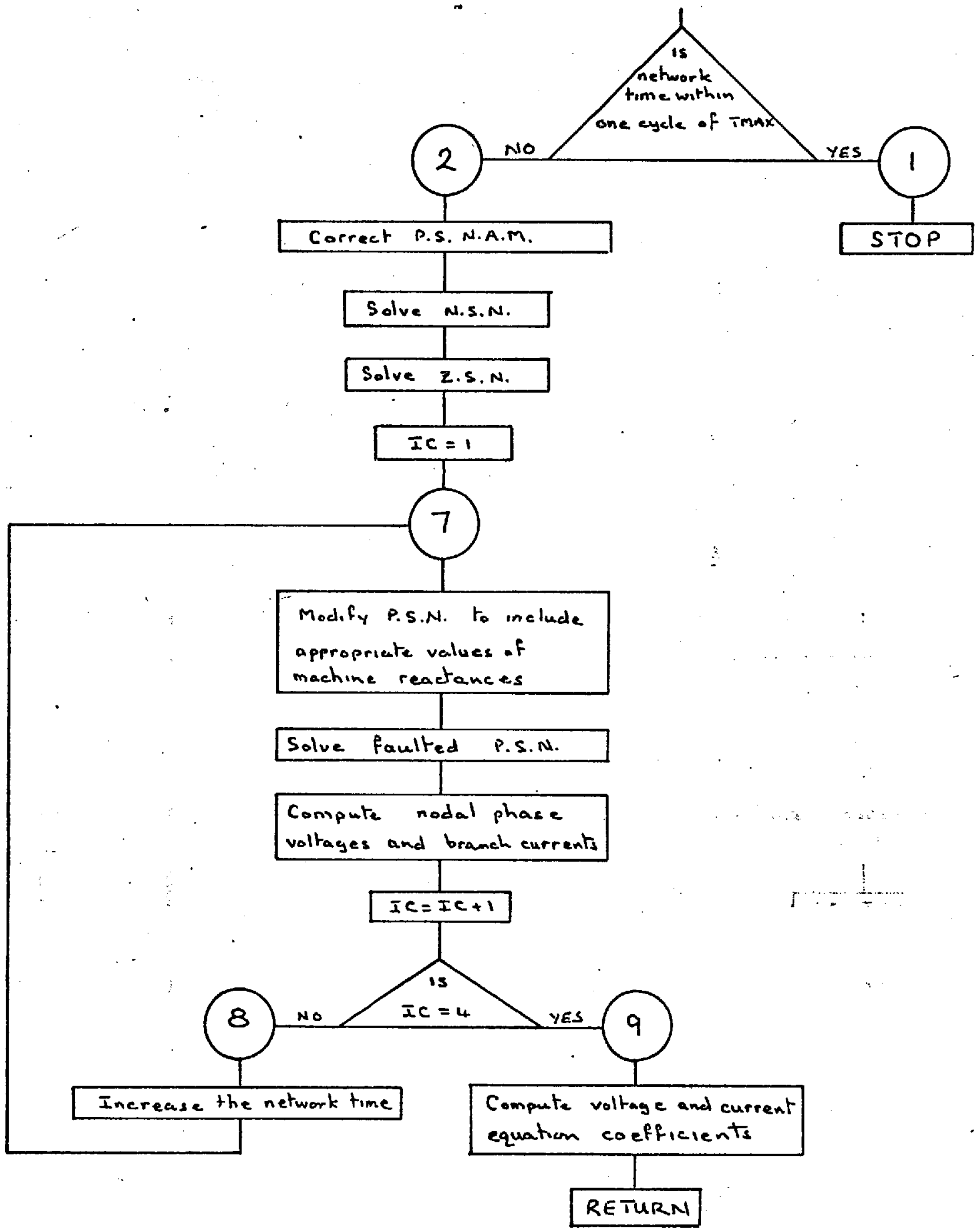
I is the number of the present iteration.

6.30 Subroutine PADE

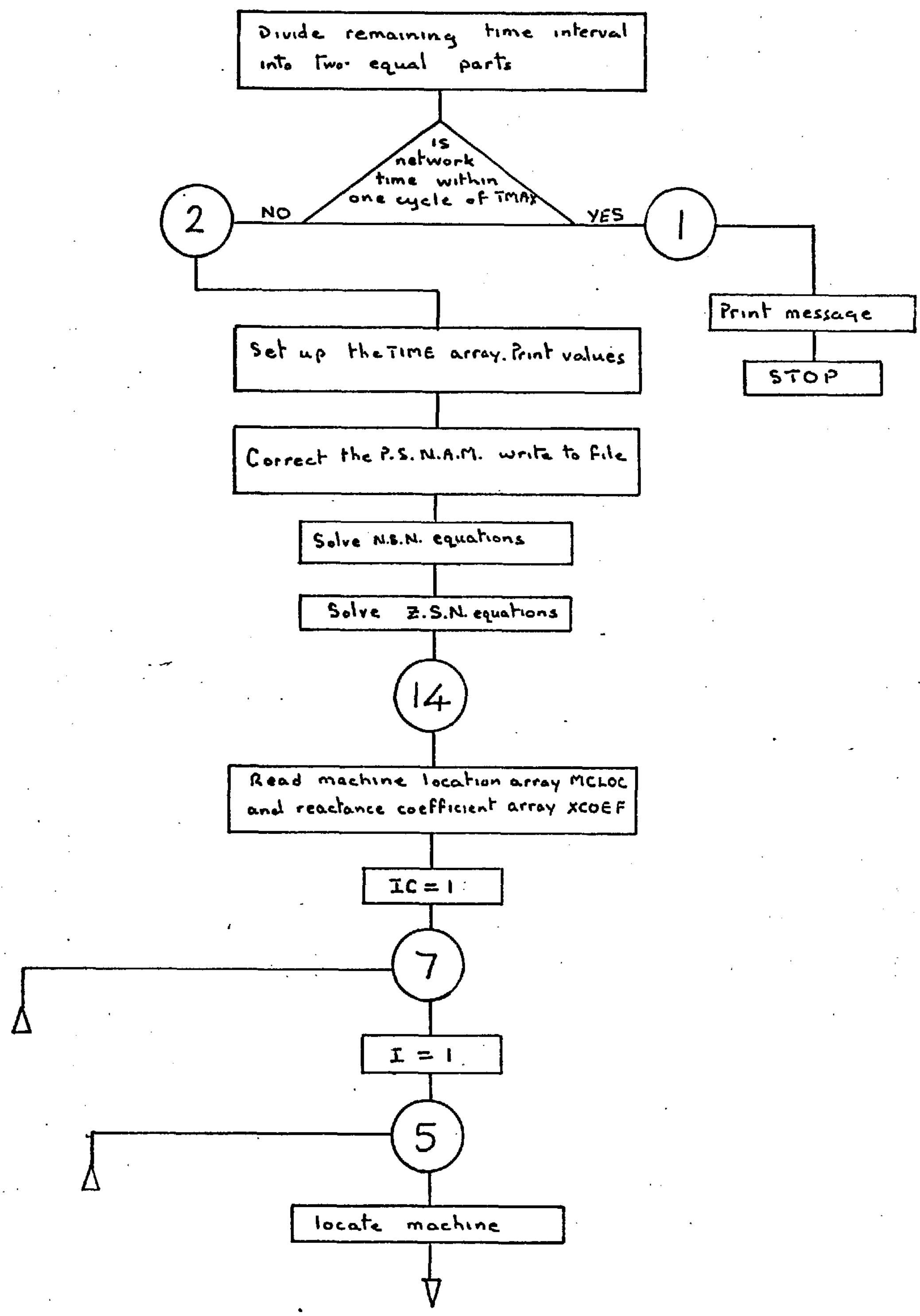
Subroutine PADE is called by ACCL1 when the five sets of nodal voltages have been stored. PADE processes these voltages to produce the accelerated value, see 2.41.

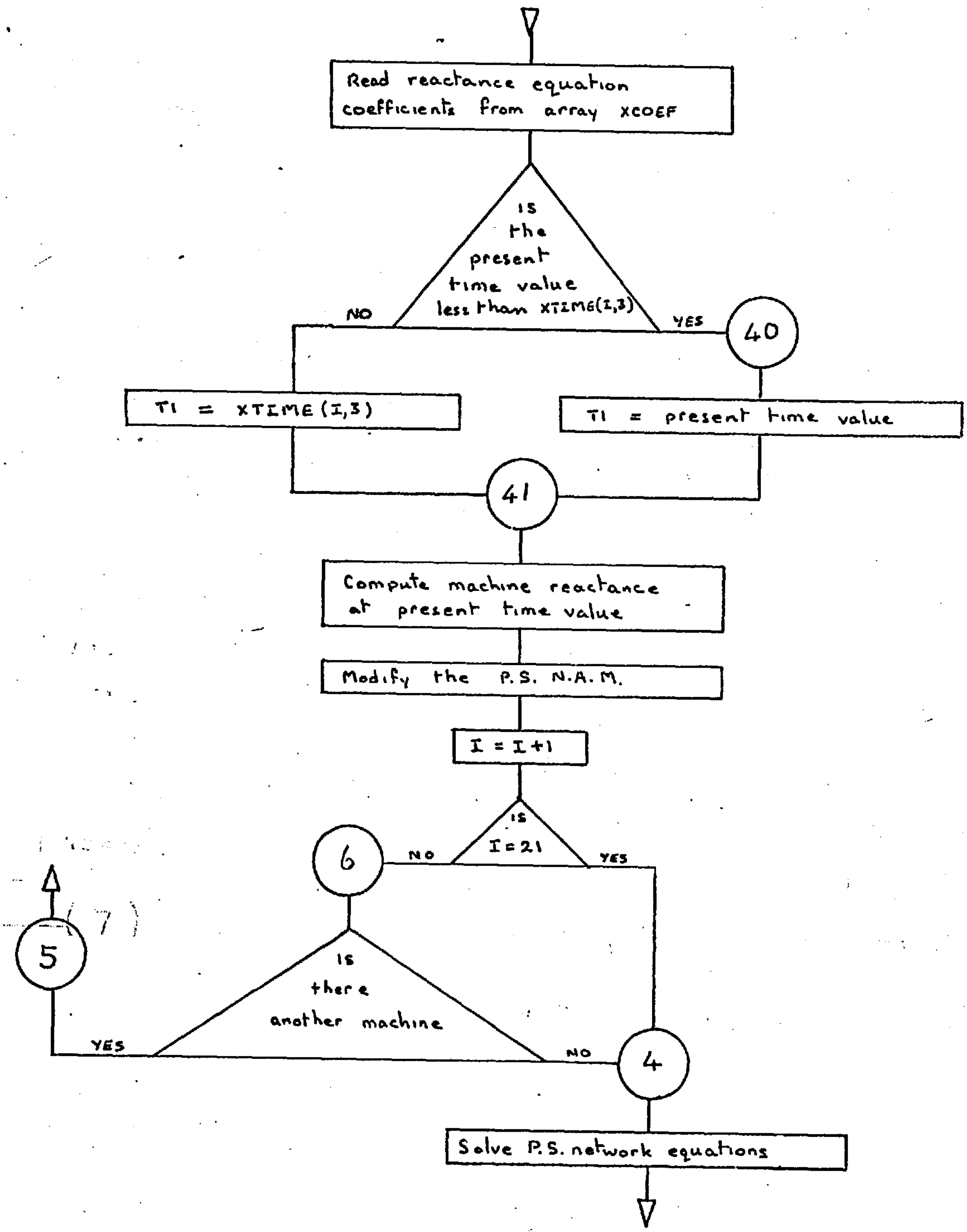
6.31

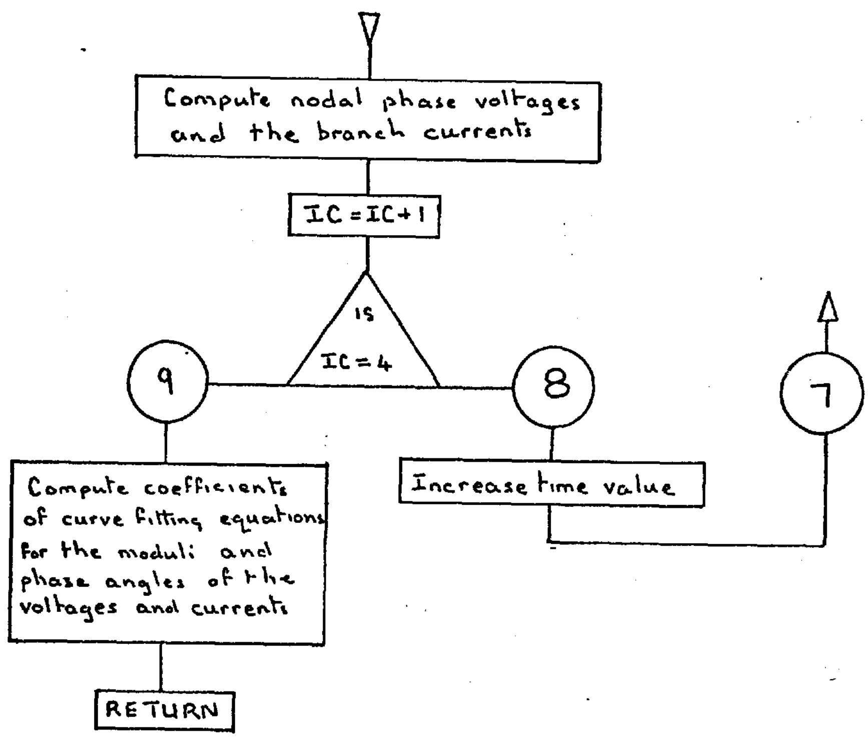
Subroutine BREAK - Macro Flow Chart



Subroutine BREAK - Flow Chart




Read reactance equation coefficients from array XCOEF
Is the present time value less than XTIME(I,3)
NO
YES
40
T1 = XTIME (I,3)
T1 = present time value
41
Compute machine reactance at present time value
Modify the P.S. N.A.M.
I = I+1
Is I=21
NO
YES
6
Is there another machine
YES
NO
5
4
Solve P.S. network equations

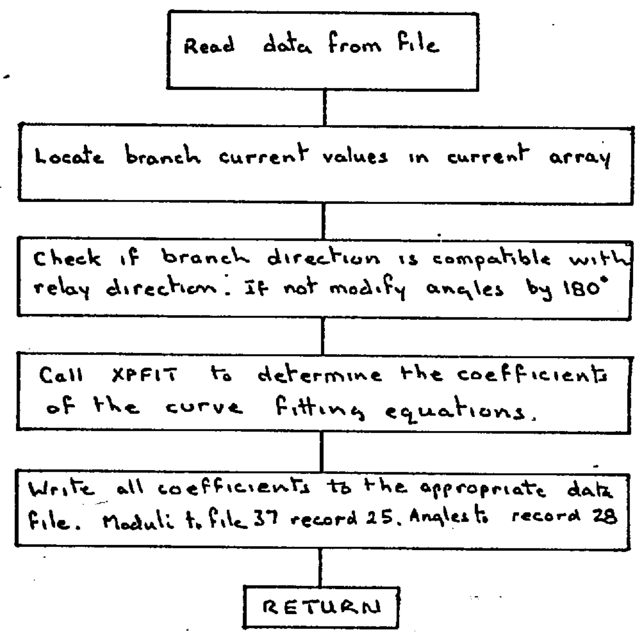


Compute nodal phase voltages and the branch currents
IC = IC+1
IS IC = 4
9
8
7
Increase time value
Compute coefficients of curve fitting equations for the moduli and phase angles of the voltages and currents
RETURN

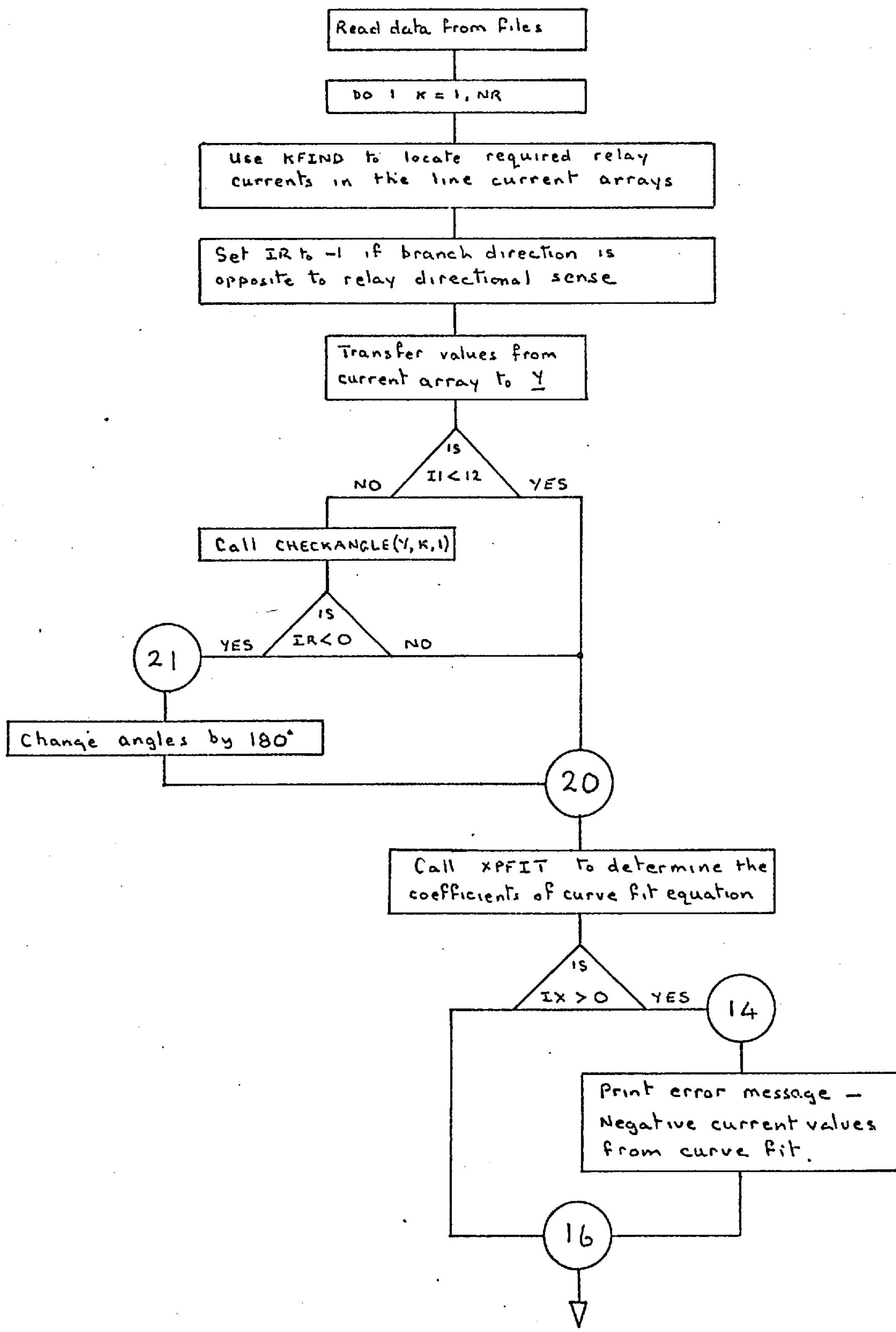
6-95.

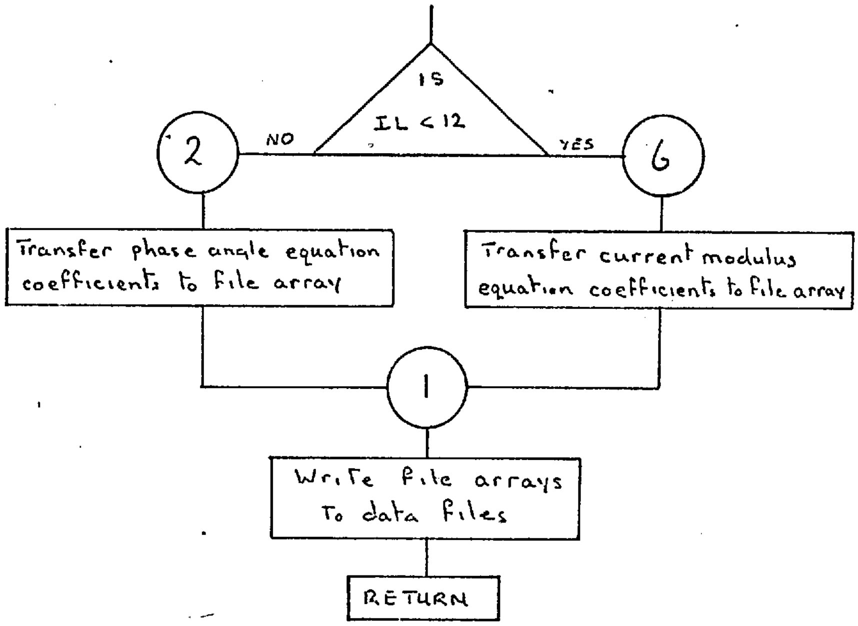
6.32

Subroutine XPEQU - Macro Flow Chart



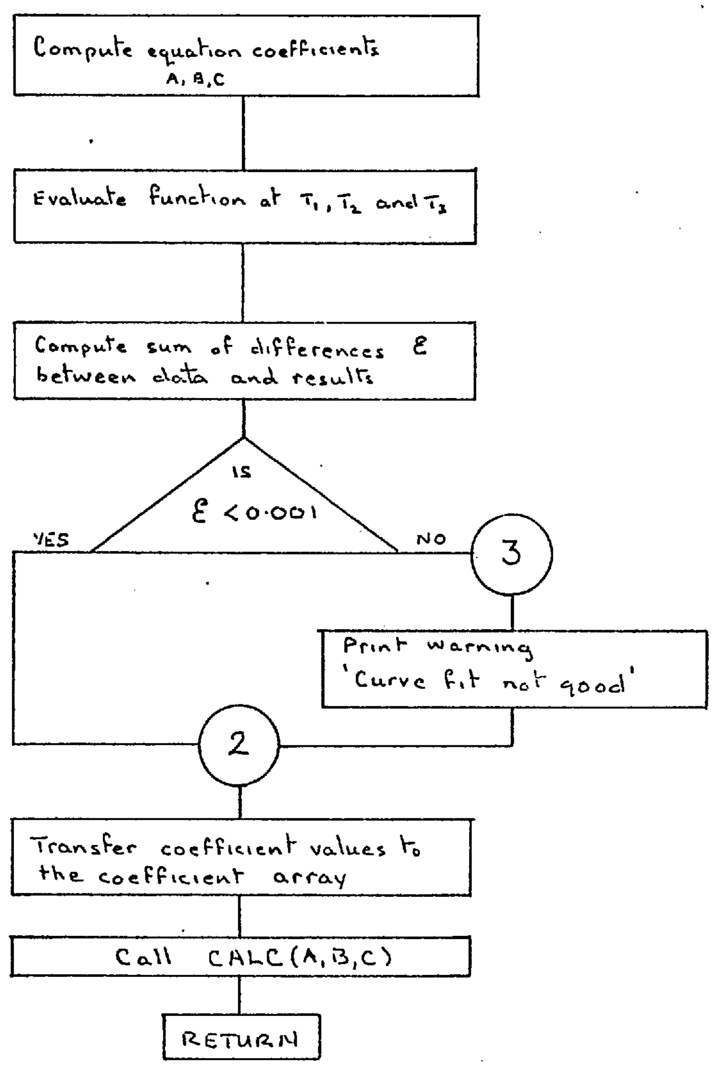
Subroutine XPEQU - Flow Chart




IS
IL < 12
NO
YES
2
6
Transfer phase angle equation coefficients to file array
Transfer current modulus equation coefficients to file array
1
Write file arrays to data files
RETURN

6.33

Subroutine XPFIT - Flow Chart



6.34 Subroutine ROTAT

Subroutine ROTAT uses the initial system data, supplied as part of the nodal data, and held in array MPS, to compute the relative angles of the nodal voltages allowing for the transformer winding connections. This is achieved by rotating the positive sequence voltage by ± 30 degrees, and the negative sequence voltage by ∓ 30 degrees depending on the transformer connections, as shown by the table below:

Fault Location	N1	Positive Sequence Voltage N2 = 1	Negative Sequence Voltage N2 = 2
⅄ side	1	N3 = N1 + N2 = 2	N3 = N1 + N2 = 3
Δ side	2	N3 = N1 + N2 = 3	N3 = N1 + N2 = 4

N3	Sequence voltage rotation
Odd	+ 30 degrees
Even	- 30 degrees

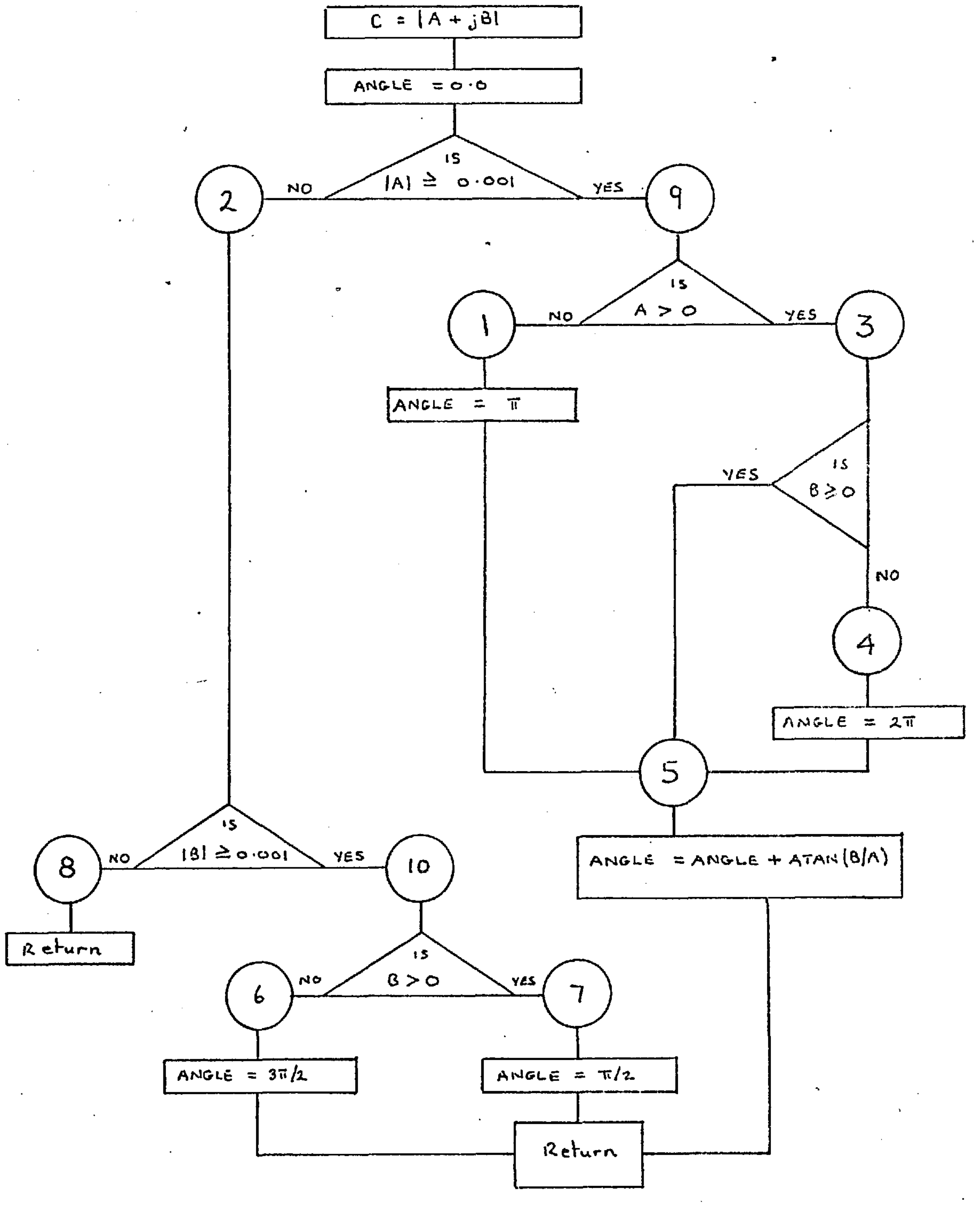
6.35 Subroutine LMT

Subroutine LMT is an ancillary routine which is used for obtaining the modulus and angle (between 0 and 360 degrees) of a complex number - quantity - expressed in cartesian form.

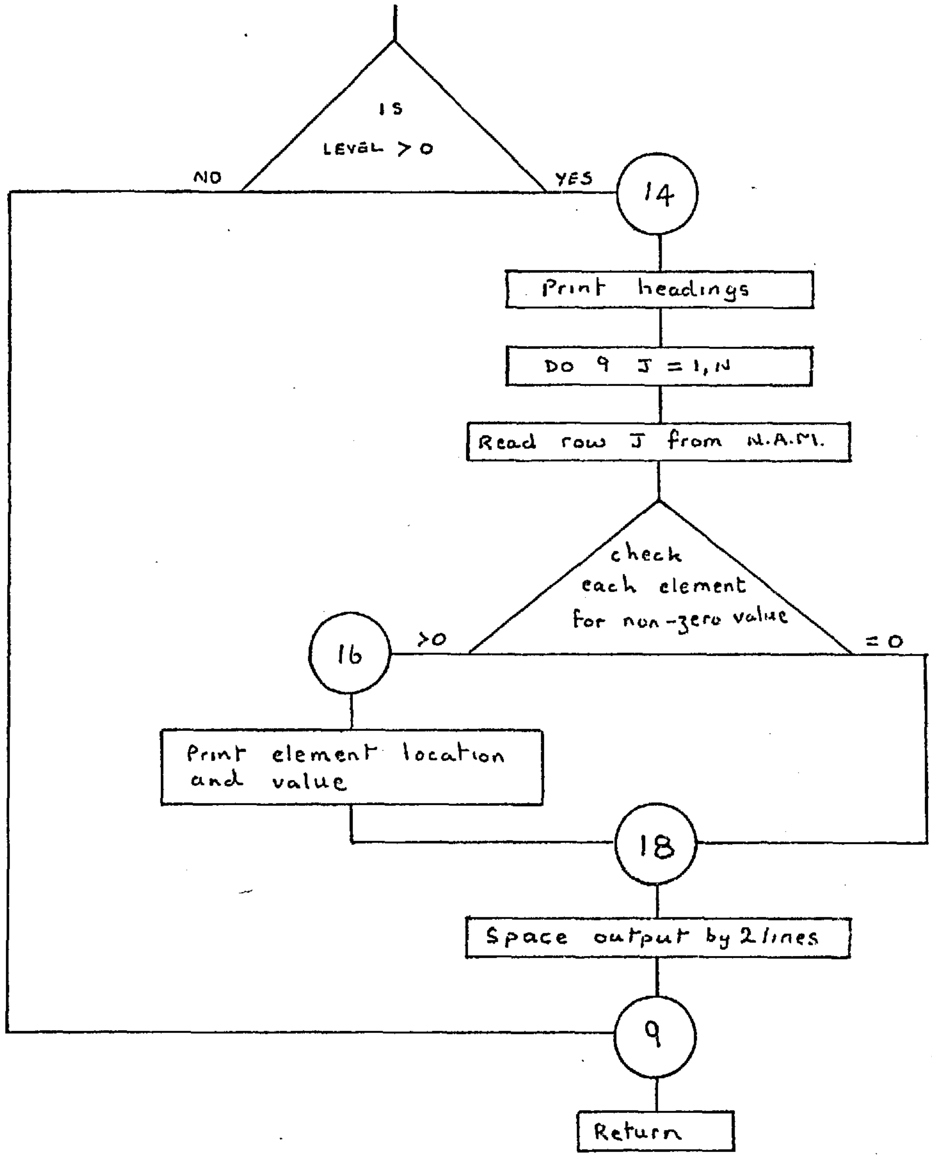
6.36 Subroutine NAM

This routine is available when trace level 1 is activated, as explained in the users manual. It prints the location and values of the non-zero elements of the nodal admittance matrices, with appropriate headings, immediately before the system equations are solved by subroutines PSA7, PSNS2 and ZSSOL.

Subroutine LMT.   Flow Chart

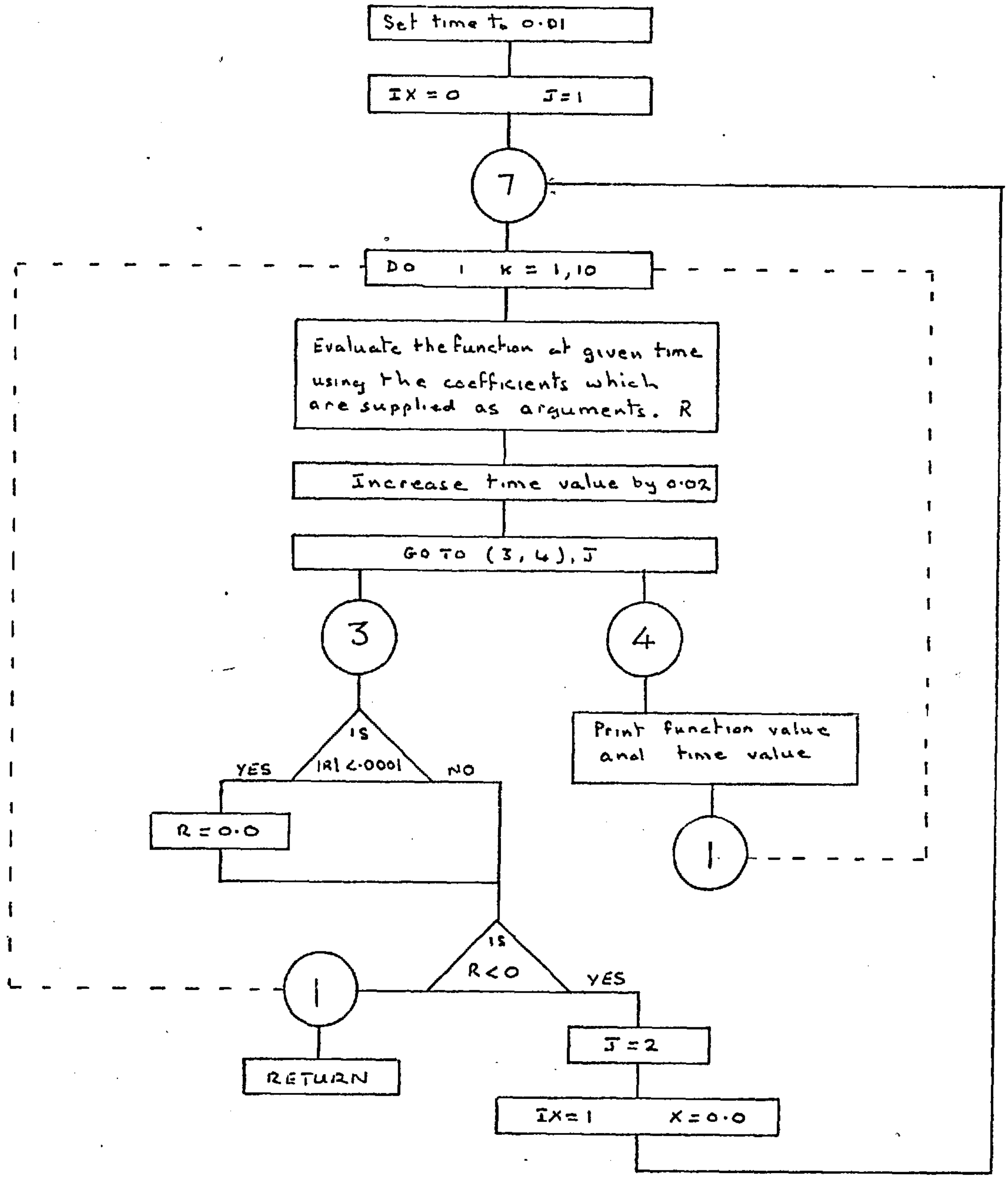


Subroutine NAM.   Flow Chart



6.37

Subroutine CALC - Flow Chart



6.38 Function MOVE(K,I)

As the network time increases, the voltages and currents decay from their maximum values to a steady state value. Thus, for some of the relays the operating current will decrease until the relays cease to move and to avoid computing a zero relay movement function Move is utilised. Function MOVE, which is used by subroutine RLMOV, inspects the no-movement array NOMOVE. Array NOMOVE has elements of value 1 or 0, 1 when a relay has made some movement during the nth time interval and zero when a relay has not moved during the nth interval. Thus, when the (n + 1)th time interval is being considered the history of each relay for the previous interval is available through function MOVE and array NOMOVE.

6.39 Function THETALIM(BL,UL,A2)

THETALIM is used to determine if the phase angle A2 lies between the specified angle limits BL and UL. BL is the lower limit and UL is the upper limit, both quoted in degrees and laying between 0 and 360. The function is used to implement the directional facility of impedance relays. It should be noted that subroutine ZRELAY, which uses THETALIM, specifies the impedance angle limits as 0 and 180 degrees.

6.40 Function THETACHK(A1, A2)

THETACHK is used to implement the directional facilityfor I.D.M.T. type relays. Since normal practise is to shift the voltage applied to the relay by -30 degrees, this function determines if the current phase angle A2 is within 30 or -90 degrees of the voltage phase angle A1. This subroutine is used in the three functions F(X), F1(X) and F2(X), which are used to evaluate the relay currents at time X.

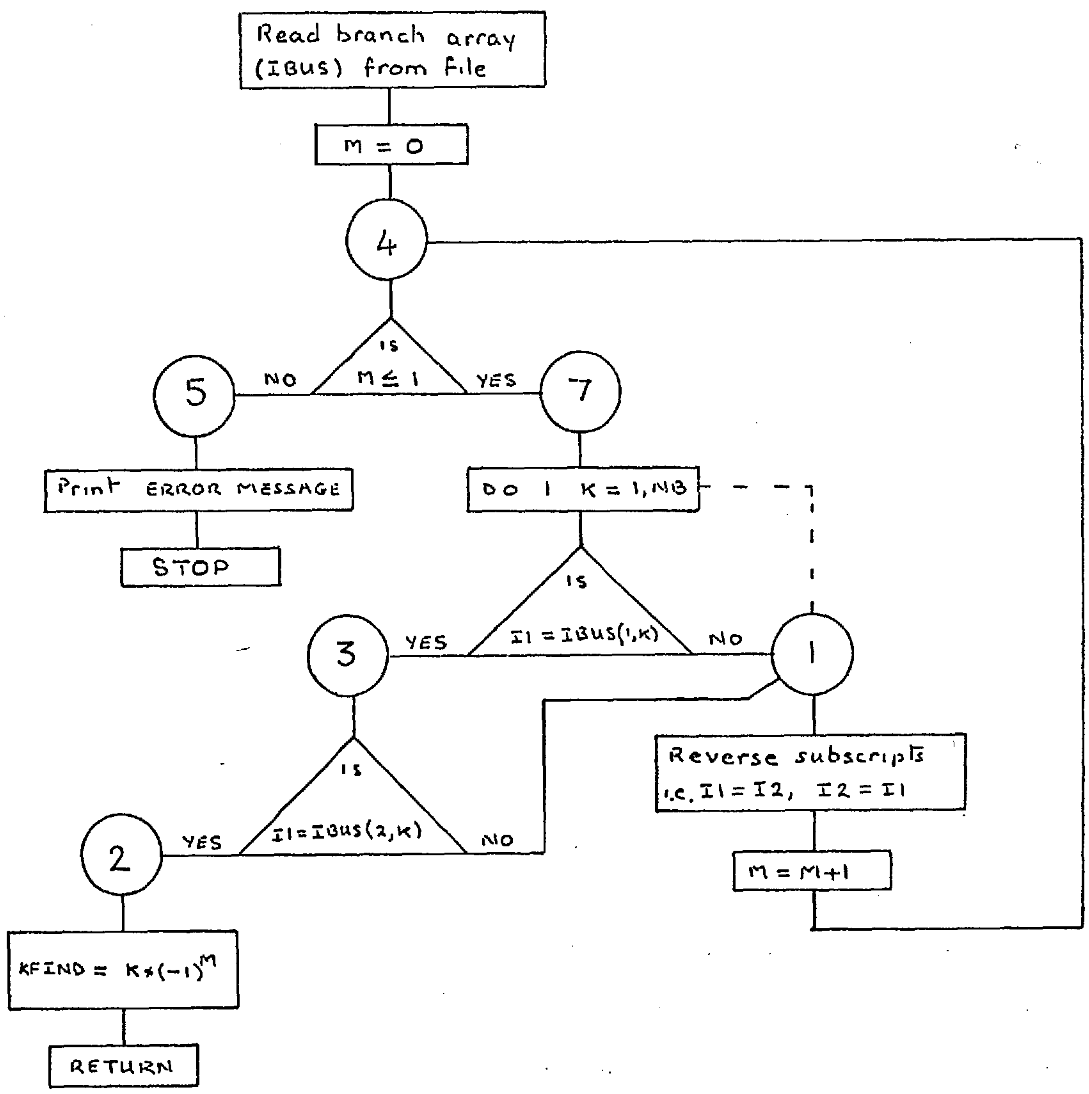
6.41 Function KFIND (I1,I2)

As described in section 6.5, the system branches are located by setting up an ordered array (branch array IBUS) of the node numbers which locate the branches. When the branch sequence and line currents are computed they are stored in arrays (also files) in an order which corresponds to that of the branch array. If a branch is defined by terminal nodes 4 and 9, say, then the computation of this branch current will take 4-9 as a positive direction. i.e. $I_{49} = (V_4 - V_9)Y_{49}$.

When a relay is specified as being located in branch K - M, KFIND(K,M) will locate the row number in which these two node numbers are stored. By specifying the relay location as K-M it is implied that the relay is situated at the node K end of the branch, KFIND will in this instance return a value of N (the row number), and -N if the node numbers were stored as M-K. This allows the phase angles of the currents to be adjusted by 180 to correspond with the relay direction.

6.42

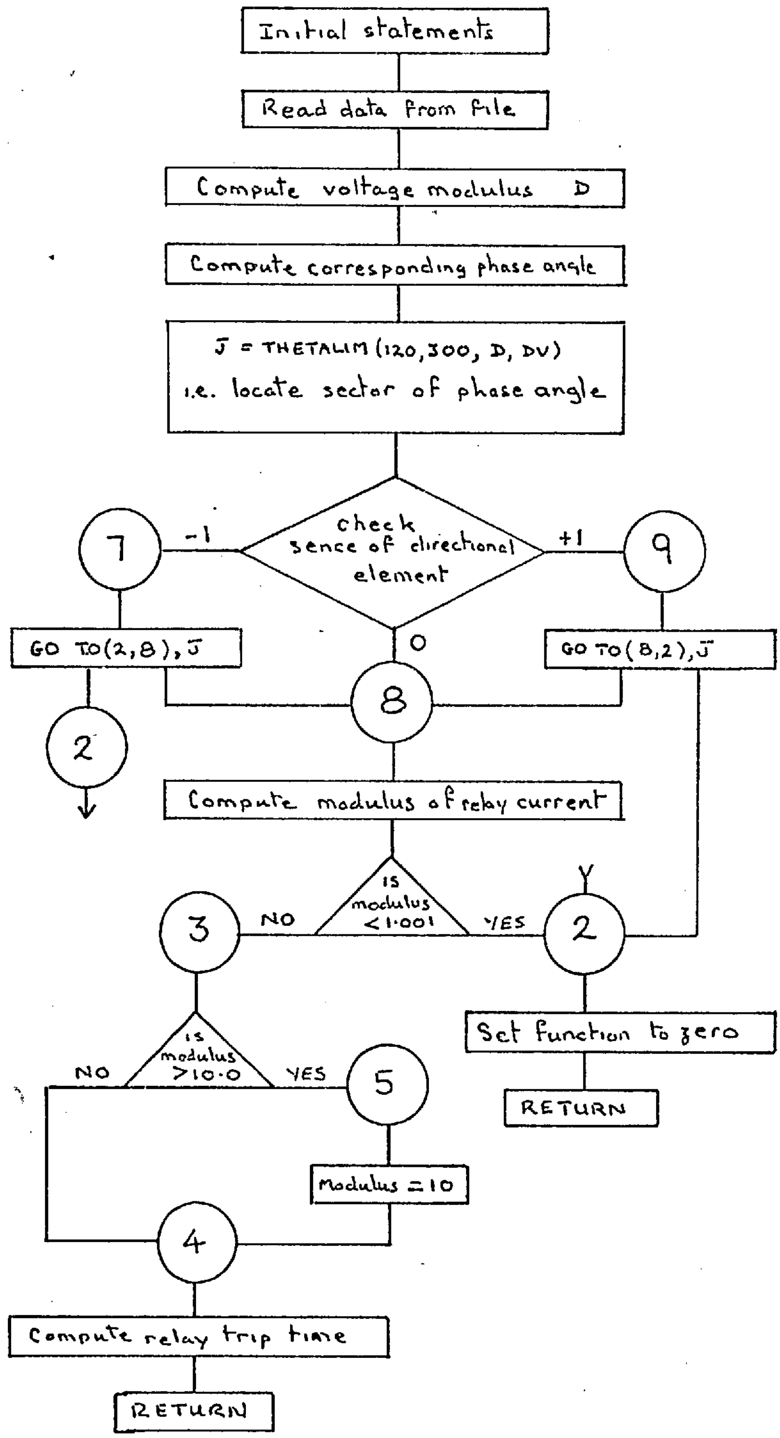
Function KFIND - Flow Chart



6.43 <u>Functions F(X), F1(X) and F2(X)</u>

These functions are used to evaluate the tripping times for the I.D.M.T. relays for the currents flowing in these relays at a time X. F1 is used for the single line to earth faults, F is used for the line to line faults and F2 is used with F for the double line to earth faults. It is necessary to use two functions for the double line to earth fault because the earth fault and the line to line fault relays will be energised at the same time.

Function F - Flow Chart



6.44 <u>The Program Trace Variable LEVEL</u>

It is possible to list the working values of several program variables by initialising LEVEL with an appropriate value. This facility has been included as a debugging aid.

a) LEVEL > 0

The non-zero elements of the sequence nodal admittance matrices are listed together with the element numbers and appropriate headings. The P.S.N.A.M. is listed by a call to NAM in subroutine PSA7. The N.S.N.A.M. is listed by a call to NAM in subroutine PSNS2, and the Z.S.N.A.M. is listed by a call to NAM in ZSSOL.

b) LEVEL > 1

From ZSSOL the values of the fault impedance, the N.S.N. impedance and the Z.S.N. impedance.

From VFCTR the values of the nodal sequence voltages, with node numbers and the headings 'Positive, Negative, Zero'.

c) LEVEL > 2

From MODS the values of the line admittances, before and after the changes.

d) LEVEL > 3

From XPFIT data values and corresponding computed values, with the message 'Results' and 'Data from XPFIT'.

From XPEQU the first ten values of the files which are listed as arguments of this routine. This achieved by a call to subroutine DAM. Also, from XPEQU a listing of the values of the arrays IBUS and IRL.

From VEQU the message 'Coefficients for the voltage equation, relay number and phase', with the appropriate values.

From BREAK a listing of the values of the arrays MCLOC, XCOEF and XTIME.

From ZSSOL the values of the effective positive sequence fault impedance.

From VFCTR the message 'IPF and IQF' and the corresponding values, where IPF + j(IQF) is the value of the current flowing from the faulted positive sequence network at the fault point.

e) LEVEL > 4

From MODS the values of the arrays MCLOC, XF, XCOEF and XTIME.

From RLMOV the values of the elements of the array TRIP as each relay is processed.

f) LEVEL > 6

From XPEQU the message 'Y from XPEQU coefficients relay number and phase', with the appropriate data values.

g) LEVEL > 8

From THETALIM the message 'Trip action inhibited by Thetalim' with the angle limits and the angle value.

h) LEVEL > 10

From RLMOV the values of the elements of the arrays CT and IRL are listed with the element numbers and a message.

i) LEVEL > 11

From BREAK the message 'K M X from BREAK', where K and M are branch location nodes and X is the value of machine reactance.

6.45 Warning and Error Messages

Subroutine Datacntrl

ERROR (1) A data termination card has not been included.

ERROR (2) The type of analysis specified by the job control card is not identifiable.

ERROR (3) The type of fault specified by the job control card is not identifiable.

ERROR (4) The job number specified is less than 1 or greater than 10.

ERROR (5) A job number has not been specified, and one or more of the following job control cards are missing:
Job Name
Data
Analysis

Subroutine RLDAT

ERROR (6) The code specifying the relay data card is not identifiable, and the data on this card is ignored. This is a non fatal error, since it is desirable to check the remaining data cards.

ERROR (7) One or both of the branch numbers specified under the IRL code are not within the range 1 to N.

Function KFIND

ERROR (8) The branch location nodes which KFIND is looking for in the IBUS array, do not exist (as a pair).

Subroutine XPEQU

ERROR (9) This message occurs when the computed curve fitting equations for the current moduli (and phase angles) generate negative values. This error results from unreasonable data values. For example, if the sequence impedances are such that the N.S. current is greater than the value of P.S. current, in corresponding sequence impedances, for one of the three analyses one of the current phase angles will be displaced from the other two by 180. Say, 330, 150, 330 - the curve fitting equations are not designed to fit this type of data, and negative values will be generated at intermediate points. The above situation also causes a similar variation in the values of the current moduli, for example 0.03, 0.0 and 0.03, results are similar to those described above. Realistic values of system data will prevent this situation from arising.

Subroutine VEQU

ERROR (10) This error occurs when negative values are generated by the curve fitting equations for the nodal voltages and their phase angles. The causes and results of this error are described under ERROR (9) above.

6.46 Miscellaneous Output

Subroutine PSA1

The input data that is required for the load flow analysis is listed.

Subroutine PSA7

Message – 'Node K is an isolated node, voltage and ND7 set to zero'. Relay operation has caused one of the system nodes to become isolated. The node is subsequently omitted from the iteration procedure.

The values of the nodal voltages are listed every 50 iterations.

Message – ' "I" iterations. These values not a solution'. The limiting number of 600 iterations has been reached, and the final values of the nodal voltages are listed. The run is terminated.

Subroutine PSNS2

The values of the nodal voltages are listed every 50 iterations.

Message – 'Fault isolated, run stopped'.

Subroutine PSAZS

Message – 'Zero sequence modifications'. The modification data values, with names, are listed.

See 5.7m.

Subroutine ZSSOL

Every 50 iterations the values of the zero sequence nodal voltages are listed.

If the number of iterations exceeds 600, the nodal voltages are listed, and the run is terminated.

6.47 The File Storage Scheme

This program has a data storage requirement in excess of 100,000 words, and in order to operate the program on a computer with 32K words of immediate access store, a random access disc backing store is required. The backing store is used by creating a series in Files, each File containing a certain number of records. Each record of a given file may be equivalent to an array, or simply the value of one program variable. This decision is made when the file is created.

The program has a general purpose common area - common area /D/, which is used to carry the values of the variables, of the active subroutine, that are not located in the other named common areas, For example, subroutine PSA1 reads data from the basic data file, file 22, processes this data and allocates it to various data files. These data files are accessed by the other subroutines as required. This is shown in diagramatic form by Fig.6.1.

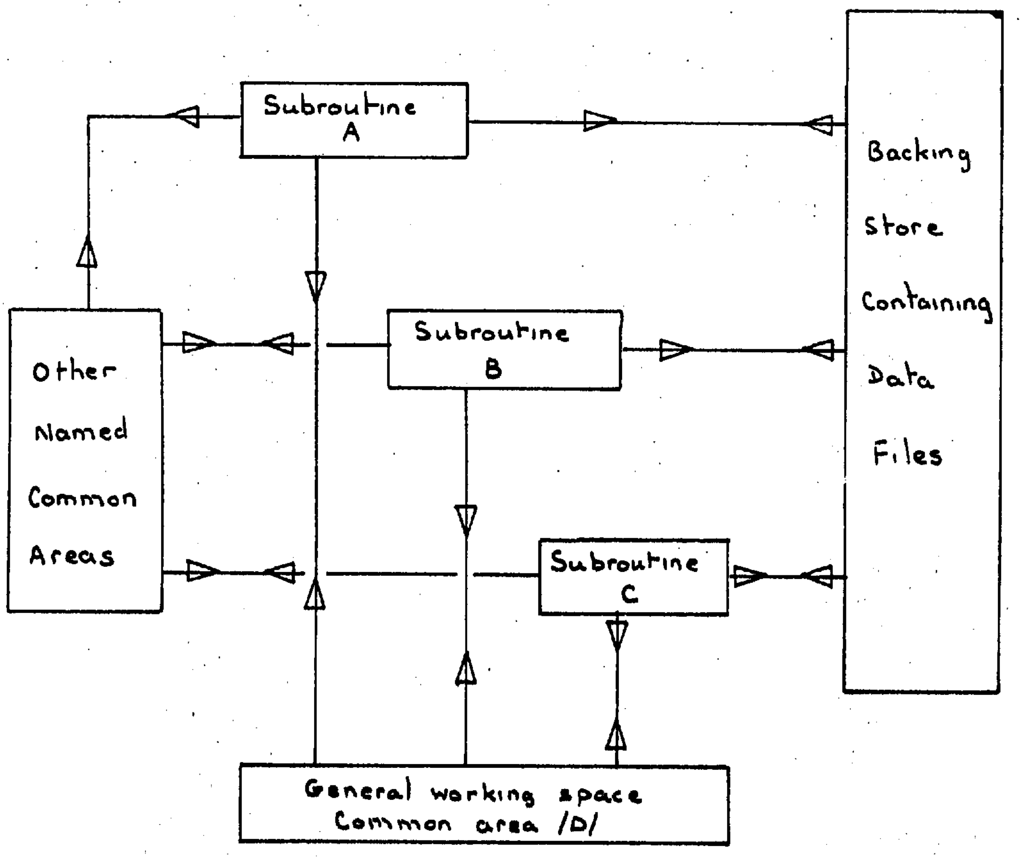

Subroutine A
Backing Store Containing Data Files
Other Named Common Areas
Subroutine B
Subroutine C
General working space
Common area /D/

Fig. 6.1

6.47.1 File Records

These files are defined by the 'Define File' statements in the master segment PSAA, see 4.2 and 6.2.

File Number	Number of Records	Words per Record	Record Number	Associated Array
1	7	240		
			1	BRI1
			2	BRI2
			3 & 4	BRR1
			5 & 6	BRR2
			7	SORT
6	80	160		G
7	80	160		B

Note: Files 6 and 7 contain the positive sequence nodal admittance matrix, where each record is one row of the matrix. The maximum number of elements in each row is 80.

10	5	160		
Injected nodal power.			1	PS
Injected nodal reactive power.			2	QS
Voltage modulus squared.			3	MODV
Initial data values of the nodal			4	ANGLE
shunt admittances, see 5.6.d.			5	YL

File Number	Number of Records	Words per Record	Description	Record Number	Associated Array
15	60	160		1	VP
				2	VQ
			Iteration starting values for	3	VP
			positive sequence voltages.	4	VQ
			Scaled values of negative	5	VNP
			sequence voltages.	6	VNQ
			Scaled values of zero	7	VZP
			sequence voltages.	8	VZQ
			Unscaled values of negative	9	VNP
			sequence voltages.	10	VNQ
			Unscaled values of zero	11	VZP
			sequence voltages.	12	VZQ
			Machine location data.	13	MCLOC
			Machine reactance data.	14	XF
			Reactance equation coefficients.	15	XCOEF
			Results of the first analysis,	16	VA
			phase A, B and C voltages.	17	VB
				18	VC
			Results of the second analysis,	19	VA
			phase A, B and C voltages.	20	VB
				21	VC
			Results of the third analysis,	22	VA
			phase A, B and C voltages.	23	VB
				24	VC

	Record Number	Associated Array
Results of the first analysis,	25	AA
phase angles for phase A, B	26	AB
and C voltages.	27	AC
Results of the second analysis,	28	AA
phase angles for phase A, B	29	AB
and C voltages.	30	AC
Results of the third analysis,	31	AA
phase angles for phase A, B	32	AB
and C voltages.	33	AC
The 'real' components of nodal	41	VP
voltages from five consecutive	42	VP
iterations, for use by the	43	VP
acceleration procedure.	44	VP
	45	VP
The quadrature components of	51	VQ
nodal voltage associated with	52	VQ
the 'real' components above.	53	VQ
	54	VQ
	55	VQ
The accelerated values of the	56	VP
nodal voltages.	57	VQ
Time constants associated with	60	XTIME
the machine reactance parameters.		

File Number	Number of Records	Words per Record	Record Number	Associated Array
19	80	160		G
20	80	160		B

Files 19 and 20, which correspond to files 6 and 7, contain the negative sequence nodal admittance matrix.

22	11 000	20		

File 22 stores system data as it is read from cards by subroutine DATACNTRL, using CREAD. The eleventh 1 000 records are reserved for temporary jobs, the other records, in groups of 1 000, are reserved for numbered jobs. Job number 1 corresponds to the first 1 000 records and job number 2 corresponds to the second 1 000 records etc. It should be noted that each job is limited to a maximum of 1 000 data cards.

23	6	80		
			1	ND
			2	ND7
			3	NDZ

Where ND7, ND and NDZ are the positive, negative and zero sequence network nodal designation matrices, respectively.

File Number	Number of Records	Words per Record	Record Number	Associated Array
24	6	480		
The positive sequence branch			1	IPP
currents.			2	IPQ
The negative sequence branch			3	INP
currents.			4	INQ
The zero sequence branch			5	IZP
currents.			6	IZQ
30	80	160		G
31	80	160		B

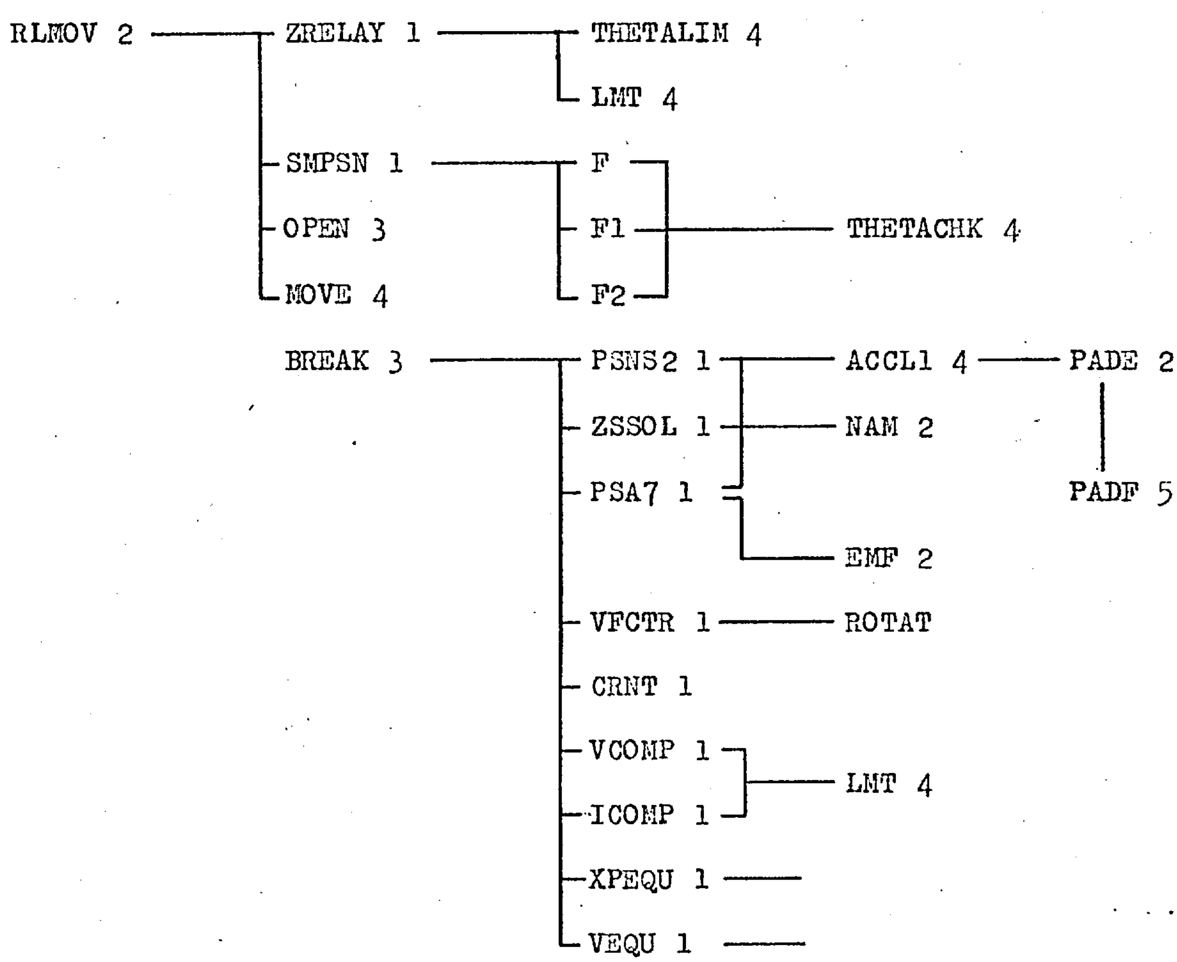
Files 30 and 31 contain the zero sequence nodal admittance matrix and thereby compliment files 6 and 7, 19 and 20.

File Number	Number of Records	Words per Record	Record Number	Associated Array
37	30	480		
The branch location array.			1 & 2	IBUS
Results of the first analysis,			3	IA
branch currents for phases A,			4	IB
B and C.			5	IC
Results of the second analysis,			6	IA
branch currents for phases A,			7	IB
B and C.			8	IC
Results of the third analysis,			9	IA
branch currents for phases A,			10	IB
B and C.			11	IC

	Record Number	Associated Array
Results of the first analysis,	12	AA
phase angles for the branch	13	BA
currents.	14	CA
Results of the second analysis,	15	AA
phase angles for the branch	16	BA
currents.	17	CA
Results of the third analysis,	18	AA
phase angles for the branch	19	BA
currents.	20	CA
Voltage angle equation coefficients.	22	VANG
See 5.6.n.2	23)) 24)	CF
See 5.6.n.2	25	CP
See 5.6.n.2	26	CT
See 5.6.n.2	27	IRL
Current angle equation coefficients.	28	ANG

6.48 <u>The Overlay Scheme</u>

The immediate access store is divided into five overlay areas and all routines are allocated to one of these. A complete listing of subroutines and their allocated areas is given in the Program Description segment, see 6.49 . The following diagram indicates the overlay relationship between major non-sequentially used routines, (the number following the subroutine name is the number of the overlay area).



```
XPEQU 1 ──────┬─ DAM          4
              ├─ XPFIT        2 ────────── CALC 4
              ├─ KFIND        2
              └─ CHECKANGLE   4

VEQU 1 ───────┬─ XPFIT        2 ────────── CALC 4
              └─ CHECKANGLE   4
```

6.49 Program Listing - Table of Contents

FORTRAN COMPILATION BY #XFAT MK 4C   DATE 13/05/75 TIME 17/20/07

```
            READ FROM(ED,FILE2(16).FORTPSAWH)
SUBFILE FORTPSAWH
            DUMP ON(PROGRAM PSAA)
            MAP
            LIBRARY(PSAB)
            LIBRARY(PSAC)
            LIBRARY(PSAB)
            LIBRARY(PSAC)
            OVERLAY PROGRAM (PSAA)
            DEPTH OF OVERLAY 8
            OVERLAY(1,1)PSA1
            OVERLAY(1,2)PSA2
            OVERLAY(1,3)PSA3
            OVERLAY(1,4)PSA4
            OVERLAY(1,5)PSA5
            OVERLAY(1,6)PSA7
            OVERLAY(1,7)PSA8
            OVERLAY(1,8)PSAZS
            OVERLAY(1,9)PSNS1
            OVERLAY(1,10)PSNS2
            OVERLAY(1,11)ZSSOL
             OVERLAY(1,12)PSA6
            OVERLAY(1,13)VCOMP
            OVERLAY(1,14)CRNT
            OVERLAY(1,15)ICOMP
            OVERLAY(1,16)VFCTR
            OVERLAY(2,1)ROTAT
            OVERLAY(2,2)RLMOV
            OVERLAY(2,3)ROT
            OVERLAY(2,4)NAM
            OVERLAY(5,1)PADF
            OVERLAY(2,6)EMF
            OVERLAY(2,7)RLDAT
            OVERLAY(2,8)XPFIT
            OVERLAY(2,10)KFIND
            OVERLAY(4,10)ACCL1
            OVERLAY(1,18)SMPSN
            OVERLAY(1,19)ZRELAY
            OVERLAY(4,12)THETACHK
            OVERLAY(3,2)OPEN
            OVERLAY(3,3)BREAK
            OVERLAY(2,5)PADE
            OVERLAY(4,7)ERROR
            OVERLAY(4,8)FREAD
            OVERLAY(4,9)CREAD
            OVERLAY(1,21)DATACNTRL
            OVERLAY(1,17)XPEQU
            OVERLAY(1,22)MODS
            OVERLAY(4,1)CALC
            OVERLAY(4,3)THETALIM
            OVERLAY(4,5)DAM
            OVERLAY(1,20)VEQU
            OVERLAY(4,4)LMT
            OVERLAY(4,2)CHECKANG
```

```
OVERLAY(4,6)MOVE
 OVERLAY(5,4)DEFBUF
OVERLAY(5,5)COMP
USE 1 = ED1/DIRECT(A1)/256
USE 10 = ED2/DIRECT(A8)/256
USE 6 = ED3/DIRECT(A2)/256
USE 7 = ED4/DIRECT(A3)/256
USE 19 = ED7/DIRECT(A4)/256
USE 20 = ED8/DIRECT(A5)/256
USE 30 = ED12/DIRECT(A6)/256
USE 31 = ED13/DIRECT(A7)/256
USE 15 = ED5/DIRECT(A9)/256
USE 23 = ED10/DIRECT(A12)/256
USE 24 = ED11/DIRECT(A14)/512
USE 37 = ED15/DIRECT(A15)/512
USE 22 = ED6/DIRECT(A10)/256
USE 29=/ARRAY
 INPUT2=CRO
 OUTPUT3=LP7
TRACE 2
 COMPRESS INTEGER AND LOGICAL
END
```

```
      MASTER PSAA
      INTEGER BB,TF,L,FAULT ,FBUS
      COMMON N,BB,L,TF,FBUS,FAULT,RN,XN,RZ,XZ
      COMMON R,X,BF,GF,ICT
      COMMON/D/XXX(1700)
      COMMON/C10/ISCIND,NCOUNT
      COMMON/C11/NODESC(20)
      COMMON/C20/I5
      COMMON/D1/IICT(10)
      DEFINE FILE15(60,80,U,N15)
      DEFINE FILE1(7,120,U,N1)
      DEFINE FILE6(80, 80,U,N6)
       DEFINE FILE7(80, 80,U,N7)
      DEFINE FILE19(80, 80,U,N19)
      DEFINE FILE20(80, 80,U,N20)
      DEFINE FILE30(80, 80,U,N30)
      DEFINE FILE31(80, 80,U,N31)
      DEFINE FILE10(5, 80,U,N10)
      DEFINE FILE23(6,40,U,N23)
      DEFINE FILE24(6,240,U,N24)
      DEFINE FILE37(30,240,U,N37)
      DEFINE FILE 22(11000,10,U,N22)
      CALL DATE(XX)
      WRITE(3,3)XX
    3 FORMAT( 1H1,///,10X,2A8, ///)
      CALL TIME(XX)
      WRITE(3,22)XX
   22 FORMAT(1H+,40X,2A8 ,///)
      CALL DATACNTRL
      CALL PSA1
       CALL PSA2
      CALL PSA3
      CALL PSA4
      CALL PSA3
      CALL PSA4
      CALL PSA5
      CALL PSA6
      IC=1
    5 CONTINUE
      CALL PSA7
      CALL PSA8
      IF(IICT(5).EQ.1)STOP
      IF(IC.EQ.0)GO TO 5
      CALL PSNS1
      IF(ISCIND)6,6,7
    7 IC = 0
      GO TO 5
    6 CONTINUE
      CALL PSNS2
      CALL PSAZS
      CALL ZSSOL
    4 CONTINUE
      CALL PSA7
      CALL VECTR
      CALL CRNT(1,6,7)
      CALL CRNT(5,19,20)
      CALL CRNT(7,30,31)
      CALL VCOMP(IC-1)
```

```
      CALL ICOMP(IC-1)
      IF(IICT(5).EQ.2)STOP
      IC=IC+1
      IF(IC-5)1,2,2
    1 CALL MODS(IC-1)
      GO TO 4
    2 CONTINUE
      CALL RLDAT
      CALL XPEQU(3,6,9,1)
      CALL XPEQU(4,7,10,2)
      CALL XPEQU(5,8,11,3)
      CALL XPEQU(12,15,18,1)
      CALL XPEQU(13,16,19,2)
      CALL XPEQU(14,17,20,3)
      CALL VEQU(16,19,22,1)
      CALL VEQU(17,20,23,2)
      CALL VEQU(18,21,24,3)
      CALL VEQU(25,28,31,1)
      CALL VEQU(26,29,32,2)
      CALL VEQU(27,30,33,3)
    8 CALL RLMOV
      CALL BREAK
      GO TO 8
      STOP
      END

D OF SEGMENT, LENGTH   554, NAME  PSAA
```

```
      SUBROUTINE PSA1
      INTEGER BB,TF,L,FAULT ,FBUS
      REAL G(80),B(80),MODV(80),ANGLE(80),YL(80)
      DIMENSION PS(80),QS(80),VP(80),VQ(80)
      DIMENSION MPS(3,80),NDZ(80),ND(80),ND7(80)
       COMMON N,BB,L,TF,FBUS,FAULT,RN,XN,RZ,XZ,R,X,BF,GF
      COMMON ICT,IEMF,TOL,LEVEL
      COMMON/D/G,B,MODV,ANGLE,YL,PS,QS,VP,VQ,MPS,NDZ,ND,ND7
      COMMON/T1/JOBNUM
      COMMON/B/A2(10)
      COMMON/D1/IC(10)
      COMMON/T2/CARD(10)
      COMMON/T3/K3
      COMMON/C10/ISCIND,NCOUNT
      COMMON/C11/NODESC(20)
      COMMON/T4/IPRINT
      CALL DEFBUF(29,80,CARD)
      K3=JOBNUM*1000+1
      READ(22'K3)IC
      K3=K3+1
      READ(22'K3)A2
      K3=K3+1
   20 FORMAT(5X,10A8)
      WRITE(3,20)A2
      WRITE(3,21)
   21 FORMAT( ///)
      NCOUNT=-1
      CALL FREAD
      READ(29,7)TOL,LEVEL,ISCIND,IPRINT
      IF(TOL.EQ.0)TOL=1.0E-6
    7 FORMAT(E20.6,3I3)
      WRITE(3,22)TOL,LEVEL,ISCIND
   22 FORMAT(///,5X,'FOR THIS STUDY',//,5X,'THE VOLTAGE'
     1 ' TOLERANCE IS',E20.6,//,5X,
     2 'THE TRACE LEVEL IS',I4,//,5X,
     3 'A SHORT CIRCUIT STUDY WILL BE MADE AT',
     4 I6,2X,'BUSBARS',///)
      IF(ISCIND)18,19,18
   18 CONTINUE
      CALL FREAD
      READ(29,23)(NODESC(J),J=1,20)
   23 FORMAT(20I3)
   19 CONTINUE
      CALL FREAD
      READ(29,111)BB,N,TF,ICONT,ISWGN,IEMF
  111 FORMAT(6I3)
      ICT=0
      DO 4 I=1,N
      G(I)=0.0
      B(I)=0.0
    4 CONTINUE
      DO 1K=1,N
      WRITE(6'K)G
      WRITE(7'K)B
    1 CONTINUE
      DO 112 K=1,N
  115 FORMAT(3I4,5F10.4)
      CALL FREAD
```

```
      READ(29,115)M,ND7(M),MPS(2,M),PS(M),QS(M)
     1,MODV(M),YL(M),ANGLE(M)
      VP(M)=MODV(M)
      MODV(M)=MODV(M)**2
      VQ(M)=0.0
      MPS(1,M)=M
  112 CONTINUE
      IF(ICONT)8,8,9
    9 CONTINUE
      CALL FREAD
      READ(29,10)K
   10 FORMAT(I3)
      DO 11 J=1,K
      CALL FREAD
      READ(29,12)M,VP(M),VQ(M)
   12 FORMAT(I3,2F10.6)
   11 ND7(M)=0
    8 CONTINUE
      DO 13 M=1,N
      IF(ND7(M))5,5,6
    6 VP(M)=0.899
      J=1
      GO TO 15
    5 J=-1
   15 ND(M)=J
      NDZ(M)=J
   13 CONTINUE
      IF(ISWGN)17,16,17
   17 ND(1)=1
      NDZ(1)=1
   16 CONTINUE
      WRITE(23'3)NDZ
      WRITE(10'1)PS
      WRITE(10'2)QS
      WRITE(10'3)MODV
      WRITE(10'4)ANGLE
      WRITE(10'5)YL
      WRITE(15'1)VP
      WRITE(15'2)VQ
      WRITE(15'3)VP
      WRITE(15'4)VQ
      WRITE(23'1)ND
      WRITE(23'4)MPS
      WRITE(23'2)ND7
      WRITE(3,24)
   24 FORMAT(6X,'P',11X,'Q',11X,'MODV**2',5X,'MOD AND',
     2 ' ANGLE OF Y',6X,'VP',
     1 10X,'VQ',5X,'ND7',2X,'MPS',2X,'J   ND NDZ')
      DO 2 J=1,N
      WRITE(3,3)PS(J),QS(J),MODV(J),YL(J),ANGLE(J)
     2,VP(J),VQ(J),ND7(J),
     1MPS(2,J),J,ND(J),NDZ(J)
    3 FORMAT(7F12.4,5I4)
    2 CONTINUE
      WRITE(3,804)BB,N,IF,ICONT,ISWGN,IEMF
  804 FORMAT(////,6I6,5X,' BB   N   IF   ICONT   ISWGN   IEMF')
      RETURN
      END
```

```
      SUBROUTINE PADE
      DIMENSION V1(80),V2(80),V3(80),V4(80).V5(80)
      COMMON/D/ V1,V2,V3,V4,V5
      COMMON N
      IMOVE = 1
      J1=41
      J2=42
      J3=43
      J4=44
      J5=45
    2 READ(15'J1)V1
      READ(15'J2)V2
      READ(15'J3)V3
      READ(15'J4)V4
      READ(15'J5)V5
      DO 1 K=1,N
      IF(ABS(V5(K)).LT.1.0E-6)GO TO 1
      X1=PADF(V1(K),V2(K))
      X2=PADF(V2(K),V3(K))
      X3=PADF(V3(K),V4(K))
      X4=PADF(V4(K),V5(K))
      X1=PADF(X1,X2)
      X2=PADF(X2,X3)
      X3=PADF(X3,X4)
      V3(K)=V4(K)+X3
      GO TO 1

      X1=PADF(X1,X2)
      X2=PADF(X2,X3)
      X4=PADF(X1,X2)
      V3(K)=V3(K)+PADF(X1,X2)
    1 CONTINUE
      GO TO(6,7),IMOVE
    6 WRITE(15'56)V3
      GO TO 8
    7 WRITE(15'57)V3
    8  IF(J1.EQ.51)GO TO 9
      J1=51
      J2=52
      J3=53
      J4=54
      J5=55
      IMOVE =2
      GO TO 2
    9 READ(15'56)V1
      READ(15'57)V2
      RETURN
      END
```

END OF SEGMENT, LENGTH   374, NAME PADE

```
      FUNCTION PADF(A,B)
      D=B-A
      IF(ABS(D).LT.1.0 E-5)GO TO 1
      PADF=1.0/D
      RETURN
    1 PADF=0.0
      RETURN
      END
```

END OF SEGMENT, LENGTH 61, NAME PADF

```
      FUNCTION F(X)
      DIMENSION ANG(20,3,3),VANG(20,3,3)
      DIMENSION CF(20,8,3),CT(20,4,3),CP(20.3,3),IRL(20,3,3)
      COMMON/C1/IRL
      COMMON/C2/CF
      COMMON/C3/CP
      COMMON/C4/CT
      COMMON/D/ANG
      COMMON/A/K
      COMMON/D/VANG
      READ(37'22)VANG
      READ(37'28)ANG
      I=2
      D=ANG(K,1,I)+ANG(K,2,I)*SORT(X)+ANG(K.3,I)*X
      DV=VANG(K,1,I)+VANG(K,2,I)*SQRT(X)+VANG(K,3,I)*X
      J=THETACHK(DV,D)
      IF(IRL(K,3,I))7,8,9
    7 GO TO(2,8),J
    9 GO TO(8,2),J
    8 CONTINUE
      A=CP(K,1,I)+CP(K,2,I)*SQRT(X)+CP(K,3,I)*X
      A=A*CT(K,2,I)*CT(K,3,I)/CT(K,1,I)
      IF(A-1.001)2,2,3
    2 F=0.0
      RETURN
    3 IF(A-10.0)4,5,5
    5 A=10.0
    4 VAL=CF(K,1,I)
      DO 1 J=2,8
    1 VAL=VAL+CF(K,J,I)*A**(J-1)
      F=VAL/CT(K,4,I)
      RETURN
      END
```

END OF SEGMENT, LENGTH   283, NAME   F

```
      FUNCTION F2(X)
      DIMENSION ANG(20,3,3),VANG(20,3,3)
      DIMENSION CF(20,8,3),CT(20,4,3),CP(20.3,3),IRL(20,3,3)
      COMMON/E/NR
      COMMON/C1/IRL
      COMMON/C2/CF
      COMMON/C3/CP
      COMMON/C4/CT
      COMMON/A/K
      COMMON/D/ANG
      COMMON/D/VANG
      READ(37'22)VANG
      READ(37'28)ANG
      D=ANG(K,1,2)+ANG(K,2,2)*SQRT(X)+ANG(K.2,3)*X
      DV=VANG(K,1,2)+VANG(K,2,2)*SQRT(X)+VANG(K,2,3)*X
      J=THETACHK(DV,D)
      IF(IRL(K,3,2))7,8,9
    7 GO TO(2,8),J
    9 GO TO(6,2),J
    8 CONTINUE
      TP=0.0
      TQ=0.0
      DO 1 I=2,3
      A= +CP(K,1,I)+CP(K,2,I)*SQRT(X)+CP(K,3,I)*X
      D= +ANG(K,1,I)+ANG(K,2,I)*SQRT(X)+ANG(K,3,I)*X
      A=A*CT(K,2,I)*CT(K,3,I)/CT(K,1,I)
      TP=TP+A*COS(D)
      TQ=TQ+A*SIN(D)
    1 CONTINUE
      A=SQRT(TP*TP+TQ*TQ)
      IF(A-1.001)2,2,3
    2 F2=0.0
      RETURN
    3 IF(A-10.0)4,5,5
    5 A=10.0
    4 VAL=CF(K,1,2)
      DO 6 J=2,8
      VAL = VAL+CF(K,J,2)*A**(J-1)
    6 CONTINUE
      F2=VAL/CT(K,4,2)
      RETURN
      END

END OF SEGMENT, LENGTH   374, NAME  F2
```



```
      FUNCTION
      DIMENSION ANG(20,3,3),VANG(20 3 3)
```

```
      FUNCTION F1(X)
      DIMENSION ANG(20,3,3),VANG(20,3,3)
      DIMENSION CF(20,8,3),CT(20,4,3),CP(20.3,3),IRL(20,3,3)
      COMMON/E/NR
      COMMON/C1/IRL
      COMMON/C2/CF
      COMMON/C3/CP
      COMMON/C4/CT
      COMMON/D/ANG
      COMMON/D/VANG
      COMMON/A/K
      READ(37'28)ANG
      READ(37'22)VANG
      D=ANG(K,1,1)+ANG(K,2,1)*SQRT(X)+ANG(K.3,1)*X
      DV=VANG(K,1,1)+VANG(K,2,1)*SORT(X)+VANG(K,3,1)*X
      J=THETACHK(DV,D)
      IF(IRL(K,3,1))7,8,9
    7 GO TO(2,8),J
    9 GO TO(8,2),J
    8 CONTINUE
      TP=0.0
      TQ=0.0
      DO 1 I=1,3
      A= +CP(K,1,I)+CP(K,2,I)*SQRT(X)+CP(K,3,I)*X
      A=A*CT(K,2,I)*CT(K,3,I)/CT(K,1,I)
      D= +ANG(K,1,I)+ANG(K,2,I)*SQRT(X)+ANG(K,3,I)*X
      TP=TP+A*COS(D)
      TQ=TQ+A*SIN(D)
    1 CONTINUE
      A=SQRT(TP*TP+TQ*TQ)
      IF(A-1.001)2,2,3
    2 F1=0.0
      RETURN
    3 IF(A-10.0)4,5,5
    5 A=10.0
    4 VAL=CF(K,1,1)
      DO 6 J=2,8
    6 VAL=VAL+CF(K,J,1)*A**(J-1)
      F1=VAL/CT(K,4,1)
      RETURN
      END
```

END OF SEGMENT, LENGTH 372, NAME F1

```
      SUBROUTINE RLDAT
      DIMENSION CF(20,8,3),CT(20,4,3),IRL(20,3,3)
      DIMENSION TIME(3),Y(3),DBARRAY(10)
      DIMENSION ZZZ(4),WW(7)
      COMMON N,NE
      COMMON/E/NR
      COMMON/G/Y,TIME
      COMMON/G/TRIP(20,3),KOUNT(20),XMAX,XMIN,TMAX
      COMMON/C1/IRL
      COMMON/C2/CF
      COMMON/C3/CP
      COMMON/C4/CT
      COMMON/T3/K3
      COMMON/T2/CARD(10)
      COMMON/D2/INTERTRIP(20,3)
      DATA ZZZ(1)/32H IRL       CT       CF1       CF2      /
      DATA WW(1)/32H A         B         C        /////////
      DATA WW(5)/24HFA        FB        FC        /
      CALL DEFBUF(29,80,CARD)
      WRITE(3,44)TMAX
   44 FORMAT(//,5X,'THIS STUDY HAS A MAXIMUM NETWORK',
     1' TIME OF',F10.2,3X,'SECONDS',///)
      CALL FREAD
      READ(29,51)NR
      WRITE(3,23)NR
   23 FORMAT(///,I10,5X,'RELAYS',///)
   51 FORMAT(I3)
      IF(NR) 100,53,100
   53 WRITE(3,54)
   54 FORMAT(5X,'NO RELAYS')
      STOP
  100 CONTINUE
      CALL FREAD
      READ(29,101)DBARRAY
  101 FORMAT(10A8)
      WRITE(3,20)DBARRAY
   20 FORMAT(10X,10A8,//)
      I2=7
      I3=1
      ITEST=0
      DO 14 K=1,7
      I=2
      CALL COMP(I,DBARRAY(1),I2,WW(K),I3)
      IF(I-2)14,15,14
   15 IF(K.GE.5)ITEST=1
      GO TO(16,17,18,11,16,17,18),K
   14 CONTINUE
      WRITE(3,9)DBARRAY
       GO TO 19
   16 L=1
      GO TO 19
   17 L=2
      GO TO 19
   18 L=3
   19 CONTINUE
      READ(29,2)J
      IF(ITEST)28,27,28
   28 READ(29,30)JJ
```

```
   30 FORMAT(8X,I3)
      IF(JJ.LT.1.OR.JJ.GT.NR)GO TO 3
   27 CONTINUE
    2 FORMAT(I2)
      J=J
      I2=3
      I3=1
      DO 3 K=1,4
      I=4
      CALL COMP(I,DBARRAY(1),I2,ZZZ(K),I3)
      IF(I-4)3,4,3
    4 GO TO(5,6,7,8),K
    3 CONTINUE
      WRITE(3,9)DBARRAY
    9 FORMAT(5X,10A8,5X,'DATA CODE ERROR')
      GO TO 100
    5 CONTINUE
      IF(ITEST)32,31,32
   32 DO 33 K=1,3
   33 IRL(J,K,L)=IRL(JJ,K,1)
      GO TO 45
   31 READ(29,10)(IRL(J,K,L),K=1,3)
   45 CONTINUE
      READ(29,1)(INTERTRIP(J,K),K=1,3)
    1 FORMAT(17X,3I3)
      IF(IRL(J,1,L).LT.1.OR.IRL(J,1,L).GT.N)GO TO 300
      IF(IRL(J,2,L).LT.1.OR.IRL(J,2,L).GT.N)GO TO 300
      GO TO 100
    6 CONTINUE
      IF(ITEST)35,34,35
   35 DO 36 K=1,4
   36 CT(J,K,L)=CT(JJ,K,1)
      GO TO 100
   34 READ(29,12)(CT(J,K,L),K=1,4)
      GO TO 100
    7 IF(ITEST)38,37,38
   38 DO 39 K=1,8
   39 CF(J,K,L)=CF(JJ,K,1)
      GO TO 100
   37 READ(29,13)(CF(J,K,L),K=1,4)
      GO TO 100
    8 IF(ITEST)40,41,40
   40 DO 42 K=5,8
   42 CF(J,K,L)=CF(JJ,K,1)
      GO TO 100
   41 READ(29,13)(CF(J,K,L),K=5,8)
      GO TO 100
   13 FORMAT(8X,4E15.6)
   10 FORMAT(8X,3I3)
   12 FORMAT(8X,4F10.4)
   11 WRITE(3,22)
   22 FORMAT(10X,'END OF RELAY DATA')
      WRITE(3,43)
   43 FORMAT(1H1)
      WRITE(37'23)CF
      WRITE(37'25)CP
      WRITE(37'26)CT
      WRITE(37'27)IRL
```

```
      RETURN
  300 WRITE(3,301)DBARRAY
  301 FORMAT(20X,10A8//,'DATA CODE ERROR - BRANCH',
     1' NUMBER OUT OF RANGE.  RUN STOPPED. '  )
      STOP
       END

END OF SEGMENT, LENGTH   700, NAME  RLDAT
```

```
      SUBROUTINE DATACNTRL
      INTEGER FBUS,FAULT
      DIMENSION DATA1(12)
      DIMENSION DATA2(6),DATA3(4),CARD(10)
      DIMENSION A(10),IC(10)
      COMMON N,NB,L,ITF,FBUS,FAULT
      COMMON/T1/JOBNUM
      COMMON/T2/CARD
      COMMON/T3/K3
      COMMON/O1/IC
      COMMON/D/ICT(10)
      COMMON/H/A
      COMMON/G/Y(3),TIME(3),TRIP(20,3),KKOUNT(20)
      COMMON/G/XMAX,XMIN,TMAX
      DATA DATA1(1)/32H/*JOB NAME      /*DATA          /
      DATA DATA1(5)/32H/*JOB NUMBER    /*END           /
      DATA DATA1(9)/32H/*ANALYSIS      /*PRINT DATA    /
      DATA DATA1(11)/16H/*PRINT DATA    /
      DATA DATA2(1)/32HLOAD FLOW       FAULT           /
      DATA DATA2(5)/16HPROTECTION      /
      DATA DATA3(1)/32HL/E     L/L     L/L/F   S/C     /
      DATA DATA4/8H////////
      CALL DEFBUF(29,80,CARD)
      DO 40 K=1,10
      A(K)=0
   40 IC(K)=0
      KOUNT=0
      NDUM=2
    1 CONTINUE
      READ(2,2)CARD
    2 FORMAT(10A8)
      WRITE(3,2)CARD
      DO 3 K=1,6
      I=8
      CALL COMP(I,CARD(1),1,DATA1(2*K-1),1)
      IF(I-8)3 ,51,3
   51 I=8
      CALL COMP(I,CARD(2),1,DATA1(2*K),1)
      IF(I-8)3,5,3
    5 IC(K)=1
      GO TO(6,1,8,9,10,1),K
    6 READ(29,2)A
      GO TO 1
    8 READ(29,7)JOBNUM
    7 FORMAT(40X,I3)
      IF(JOBNUM.LT.0.OR.JOBNUM.GT.10)CALL ERROR(4)
      GO TO 1
   10 DO 11 J=1,3
      I=8
      CALL COMP(I,CARD(3),1,DATA2(2*J-1),1)
      IF(I-8)11,50,11
   50 I=8
      CALL COMP(I,CARD(4),1,DATA2(2*J),1)
      IF(I- 8)11,12,11
   12 IC(5)=J
      IF(J.EQ.1)GO TO 1
      DO 13 J=1,4
      I=5
```

```
      CALL COMP(I,CARD(7),1,DATA3(J),1)
      IF(I-5)13,41,13
   41 IC(8)=J
      FAULT=J
      READ(29,141)TMAX
  141 FORMAT(65X,F7.0)
      READ(29,140)FBUS
  140 FORMAT(60X,I3)
      IC(7)=FBUS
      GO TO 1
   13 CONTINUE
      CALL ERROR(3)
   11 CONTINUE
      CALL ERROR(2)
    9 CONTINUE
      IF(IC(3).NE.0)GO TO 110
      IF(IC(1).EQ.0)GO TO 111
      IF(IC(2).EQ.0)GO TO 111
      IF(IC(5).EQ.0) GO TO 111
      WRITE(22'1)IC
      WRITE(22'2)A
      K3=2
      GO TO 121
  111 CALL ERROR(5)
  110 K3=JOBNUM*1000+1
      READ(22'K3)ICT
      K3=K3+1
      IF(IC(1).EQ.0)GO TO 117
      WRITE(22'K3)A
      GO TO 118
  117 READ(22'K3)A
  118 IF(IC(5).EQ.0)GO TO 119
      ICT(5)=IC(5)
      ICT(7)=IC(7)
      ICT(8)=IC(8)
      WRITE(22'K3-1)ICT
  119 CONTINUE
      FAULT=ICT(8)
      FBUS=ICT(7)
      IF(IC(2).EQ.0)RETURN
  121 K3=K3+1
  300 CONTINUE
      CALL CREAD
      I=8
      CALL COMP(I,CARD(1),1,DATA4,1)
      IF(I.NE.8)GO TO 300
      RETURN
    3 CONTINUE
      CALL ERROR(1)
      RETURN
      END
```

END OF SEGMENT, LENGTH   742, NAME DATACNTRL

```
      SUBROUTINE CREAD
      COMMON/T3/K
      COMMON/T2/CARD(10)
      READ(2,1)CARD
      WRITE(22'K)CARD
    1 FORMAT(10A8)
      K=K+1
      RETURN
      END
```

END OF SEGMENT, LENGTH   41, NAME CREAD

```
      SUBROUTINE FREAD
      COMMON/T2/CARD(10)
      COMMON/T3/K
      COMMON/D1/IC(10)
      READ(22'K)CARD
      IF(IC(6).EQ.0)GO TO 3
      WRITE(3,2)K
    2 FORMAT(10X,I10,'     RECORD NUMBER')
      WRITE(3,1)CARD
    1 FORMAT(25X,10A8,//)
    3 K=K+1
      RETURN
      END
```

END OF SEGMENT, LENGTH   64, NAME FREAD

```
      SUBROUTINE ERROR(K)
      WRITE(3,1)K
    1 FORMAT(5X,'ERROR',I6,5X,' RUN STOPPED')
      STOP
      END

END OF SEGMENT, LENGTH     29, NAME  ERROR
```

FORTRAN COMPILATION BY #XFAT MK 4C   DATE 13/05/75 TIME 17/11/19

```
            READ FROM(ED,FILE5(12).FORTPSAWH)
SUBFILE FORTPSAWH
            SEND TO(PSAC)
            OVERLAY SEGMENTS
            TRACE 2
            COMPRESS INTEGER AND LOGICAL
            END
```

```
      FUNCTION   MOVE( K,I)
      COMMON/C6/NOMOVE(20,3)
      COMMON/C7/MCHK
      IF(MCHK)1,1,2
    2 IF(NOMOVE(K,I).EQ.0) GO TO 3
    1   MOVE=1
      RETURN
    3   MOVE= 2
      RETURN
      END

ND OF SEGMENT, LENGTH    61, NAME  MOVE
```

```
      SUBROUTINE CHECKANG(Y,K,I1)
      DIMENSION Y(3)
      I=0
      PI2=8.0*ATAN(1.0)
      A1=PI2*355.0/360.0
      A2=PI2*5.0/360.0
      DO 1 M=1,3
      IF(A1.LE.Y(M).AND.PI2.GE.Y(M))I=I+1
    1 CONTINUE
      IF(I.EQ.0.OR.I.EQ.3) GO TO 3
      IF(I.EQ.1)GO TO 2
      DO 4 M=1,3
      IF(Y(M)-A2)30,30,4
   30 Y(M)=PI2
      I2=1
      GO TO 20
    4 CONTINUE
    3 RETURN
    2 DO 6 M=1,3
      IF(Y(M)-A1)6,31,31
   31 Y(M)=0,0
      I2=2
      GO TO 20
    6 CONTINUE
      RETURN
   20 WRITE(3,7)K
    7 FORMAT(5X,'RELAY POSITION',I4 )
      GO TO (9,10),I1
    9 WRITE(3,8)
    8 FORMAT(5X,'CURRENT')
      GO TO 12
   10 WRITE(3,11)
   11 FORMAT(5X,'VOLTAGE')
   12 WRITE(3,13)
   13 FORMAT(1H+,14X,'ANGLE ADJUSTED TO')
      GO TO(14,15),I2
   14 WRITE(3,16)
   16 FORMAT(1H+,32X,'360.0 DEGREES')
      GO TO 4
   15 WRITE(3,17)
   17 FORMAT(1H+,32X,' ZERO DEGREES')
      GO TO 6
      RETURN
      END

END OF SEGMENT, LENGTH   270, NAME  CHECKANG
```

```
      SUBROUTINE MODS(IND)
      DIMENSION MCLOC(20,2),XF(20,3),XCOEF(20,3)
      DIMENSION G(80),B(80)
      COMMON XC(13),LEVEL
      COMMON/0/G,B,MCLOC,XF,XCOEF
      COMMON/D/XTIME(20,3)
      COMMON/T3/K3
      COMMON/T2/CARD(10)
      COMMON/G/Y(3),TIME(3)
      COMMON/D/CPT(3)
      CALL DEFBUF(29,80,CARD)
      IJ=0
      WRITE(3,1)
    1 FORMAT(////,5X,'POSITIVE SEQUENCE NETWORK MODIFICATIONS')
      IF(IND-1)20,20,21
   20 DO 22 K=1,20
      DO 22 J=1,2
   22 MCLOC(K,J)=0
      DO 23 K=1,20
      DO 23 J=1,3
      XF(K,J)=0.0
      XTIME(K,J)=0.0
   23 XCOEF(K,J)=0.0
    7 IJ=IJ+1
      CALL FREAD
      READ(29,5)K,M,X1,X2,X3,TX1,TX2,TX3
    5 FORMAT(2I4,6F10.4)
      WRITE(3,4)K,M,X1,X2,X3,TX1,TX2,TX3
    4 FORMAT(5X,'REACTANCE AND TIME VALUES',2I4,6F10.4,///)
      XTIME(IJ,1)=TX1
      XTIME(IJ,2)=TX2
      XTIME(IJ,3)=TX3
      MCLOC(IJ,1)=K
      MCLOC(IJ,2)=M
      IF(K.EQ.0)GO TO 6
      DO 53 J6=1,3
   53 TIME(J6)=XTIME(IJ,J6)
      Y(1)=X1
      Y(2)=X2
      Y(3)=X3
      CALL XPFIT(CPT)
      DO 54 J6=1,3
      XF(IJ,J6)=CPT(1)+CPT(2)*SQRT(XTIME(1,J6))
     1 +CPT(3)*XTIME(1,J6)
   54 XCOEF(IJ,J6)=CPT(J6)
      GO TO 7
   21 READ(15'13)MCLOC
      READ(15'60)XTIME
      READ(15'15)XCOEF
      READ(15'14)XF
    6 DO 57 J6=1,20
      J=0
      K=MCLOC(J6,1)
      IF(K.EQ.0)GO TO 55
      M=MCLOC(J6,2)
  200 READ(7'K)B
      READ(6'K)G
      IF(LEVEL-2)9,9,11
```

```
   11 CONTINUE
      WRITE(3,10)G(K),B(K)
    9 CONTINUE
      B(K)=B(K)+B(M)-1.0/XF(J6,IND)
      IF(LEVEL-2)12,12,13
   13 CONTINUE
      WRITE(3,10)G(K),B(K)
   12 CONTINUE
      B(M)=1.0/XF(J6,IND)
      IF(LEVEL-2)14,14,15
   15 CONTINUE
      WRITE(3,10)G(M),B(M)
   14 CONTINUE
   10 FORMAT(2E20.6,5X,'G AND B FROM MODS')
      WRITE(7'K)B
      IF(J)3,3,57
    3 J=1
      KT=M
      M=K
      K=KT
      GO TO 200
   57 CONTINUE
   55 IF(IND.NE.1)GO TO 26
      WRITE(15'60)XTIME
      WRITE(15'15)XCOEF
      WRITE(15'13)MCLOC
      WRITE(15'14)XF
   26 IF(LEVEL-4)27,8,8
    8 CONTINUE
      WRITE(3,24)((MCLOC(K,J),J=1,2),K=1,20)
   24 FORMAT(2I10,5X,'MCLOC K J')
      WRITE(3,25)((XF(K,J),J=1,3),K=1,20)
   25 FORMAT(3F15.4,5X,'XF   K   J')
      WRITE(3,28)((XCOEF(K,J),J=1,3)K=1,20)
   28 FORMAT(3E20.5,5X,'XCOEF   K   J')
      WRITE(3,38)((XTIME(K,J),J=1,3),K=1,20)
   38 FORMAT(3E20.6,5X,'XTIME    K    J')
   27 CONTINUE
      READ(15'60)XTIME
      DO 16 K=1,3
   16 TIME(K)=XTIME(1,K)
      RETURN
       END

END OF SEGMENT, LENGTH   803, NAME  MODS
```

```
      SUBROUTINE ACCL1(IB,IX,I)
      DIMENSION  V1(80),V2(80),V3(80),V4(80),V5(80)
      COMMON N
      COMMON/D/V1,V2,V3,V4,V5
      IF(I-IB)1,2,2
    2 IX=IX+1
      IF(IX.EQ.6)GO TO 3
      WRITE(15'IX+40)V1
      WRITE(15'IX+50)V2
      RETURN
    3 CALL PADE
      A=15
      CALL SSWTCH(10,J10)
      IF(J10.NE.1)GO TO 4
      A=20
      GO TO 6
    4 CALL SSWTCH(11,J11)
      IF(J11.NE.1)GO TO 6
      A=25
    6 IB=IB+A
      IX=0
    1 RETURN
      END

END OF SEGMENT, LENGTH   157, NAME  ACCL1
```

```
      FUNCTION KFIND(I1,I2)
      DIMENSION IBUS(2,240)
      COMMON N,NB
      COMMON/C1/IRL
      READ(37'1)IBUS
      M=0
    4 IF(M-1)7,7,5
    7 DO 1 K=1,NB
      IF(I1-IBUS(1,K))1,3,1
    3 IF(I2-IBUS(2,K))1,2,1
    1 CONTINUE
      IT=I1
      I1=I2
      I2=IT
      M=M+1
      GO TO 4
    2 IF(M)8,8,9
    8 KFIND=K
      RETURN
    9 KFIND=-K
      RETURN
    5 WRITE(3,6)I1,I2
    6 FORMAT(5X,'KFIND IS LOOKING FOR A BRANCH',
     1' BETWEEN NODES',I4,' AND',I4,
     2 ' DATA ERROR RUN STOPPED')
      CALL ERROR(8)
      STOP
      END

END OF SEGMENT, LENGTH   145, NAME  KFIND
```

```
      SUBROUTINE ZRELAY(S)
      INTEGER FAULT
      DIMENSION VCP(20,3,3),VANG(20,3,3),CF(20,8,3)
      DIMENSION ANG(20,3,3),IRL(20,3,3),CP(20,3,3)
      DIMENSION CT(20,4,3)
      COMMON/Q/IHOLD
      COMMON/A/K
      COMMON/G/Y(3),X(3),TRIP(20,3),KOUNT(20)
      COMMON/G/XMAX,XMIN,TMAX
      COMMON NZ(5),FAULT,XC(10),LEVEL
      COMMON/C1/IRL
      COMMON/D/VCP,VANG,ANG
      COMMON/C2/CF
      COMMON/C3/CP
      COMMON/C4/CT
      READ(37'22)VANG
      READ(37'21)VCP
      READ(37'28)ANG
      WRITE(3,26)K
   26 FORMAT(5X,'IMPEDANCE RELAY NUMBER',I5)
      PI2=8.0*ATAN(1.0)
      IHOLD=0
      S=0.0
      SX=SQRT(XMAX)
      IF(FAULT-2)11,12,12
   12 CONTINUE
      VB=VCP(K,1,2)+VCP(K,2,2)*SX+VCP(K,3,2)*XMAX
      VBA=VANG(K,1,2)+VANG(K,2,2)*SX + VANG(K,3.2)*XMAX
      VC=VCP(K,1,3)+VCP(K,2,3)*SX+VCP(K,3,3)*XMAX
      VCA=VANG(K,1,3)+VANG(K,2,3)*SX+VANG(K.3,3)*XMAX
      GO TO 14
   11 VA=VCP(K,1,1)+VCP(K,2,1)*SX + VCP(K,3-1)*XMAX
      VAA=VANG(K,1,1)+VANG(K,2,1)*SX+ VANG(K,3,1)*XMAX
      VP=VA*COS(VAA)
      VQ=VA*SIN(VAA)
      IF(FAULT.EQ.1)GO TO 13
   14 VP1=VB*COS(VBA)
      VQ1=VB*SIN(VBA)
      VP2=VC*COS(VCA)
      VQ2=VC*SIN(VCA)
      VP=VP1-VP2
      VQ=VQ1-VQ2
C      VB - VC
C      L/L/E     FAULT TREATED AS  L/L
   13 CALL LMT(VP,VQ,A,R)
      A2=CP(K,1,1)+CP(K,2,1)*SX+CP(K,3,1)*XMAX
      AA=ANG(K,1,1)+ANG(K,2,1)*SX +ANG(K,3,1)*XMAX
      AB=CP(K,1,2)+CP(K,2,2)*SX +CP(K,3,2)*XMAX
      ABA=ANG(K,1,2)+ANG(K,2,2)*SX +ANG(K,3-2)*XMAX
      AC=CP(K,1,3)+CP(K,2,3)*SX+CP(K,3,3)*XMAX
      ACA=ANG(K,1,3)+ANG(K,2,3)*SX+ANG(K,3,3)*XMAX
      AP=A2*COS(AA)
      AQ=A2*SIN(AA)
      BQ=  AB*SIN(ABA)
      BP=  AB*COS(ABA)
      CAP=AC*COS(ACA)
      CAQ=AC*SIN(ACA)
      IF(FAULT-2)15,16,16
```

```
   16 AP=BP-CAP
      AQ=BQ-CAQ
C       IB  -  IC
      CALL LMT(AP,AQ,A1,R1)
      ZM=(A/A1*CT(K,1,2))/(CT(K,2,2)*CT(K,3.2))
      ZA= R-R1
      IF(ZA)27,28,28
   27 ZA=PI2+ZA
   28 CONTINUE
      IF(IRL(K,3,2)-2)5,5,6
    5 J=THETALIM(0.0,180.0,ZA)
      GO TO 9
    6 J=THETALIM(180.0,360.0,ZA)
    9 GO TO(8,21),J
    8 DO 10 M=1,3
      IF(ZM.GT.CF(K,2*M-1,2))GO TO 10
      IF(XMAX.LT.CF(K,2*M,2))GO TO 10
      S=1.001
      GO TO 21
   10 CONTINUE
      GO TO 21
   15 CONTINUE
       AC=CP(K,1,3)+CP(K,2,3)*SX +CP(K,3,3)* XMAX
       ACA=ANG(K,1,3)+ANG(K,2,3)*SX+ANG(K,3.3)*XMAX
       CCP=AC*COS(ACA)
       CCQ=AC*SIN(ACA)
      BQ=AQ+(AQ+BQ+CCQ)/3.0
      BP=BP+(AP+BP+CCP)/3.0
      CALL LMT(BP,BQ,A1,R1)
C     Z0=2*Z1   HENCE (K-1)/3 = 1/3
      ZM= (A/A1)*CT(K,2,2)*CT(K,3,2)/CT(K,1.2)
      ZA= R-R1
      IF(ZA)29,30,30
   29 ZA=PI2+ZA
   30 CONTINUE
      IF(IRL(K,3,1)-2)25,25,17
   25 J=THETALIM(0.0,180.0,ZA)
      GO TO 18
   17 J=THETALIM(180.0,360.0,ZA)
   18 GO TO(19,21),J
   19 DO 20 M=1,3
      IF(ZM.GT.CF(K,2*M-1,1))GO TO 20
      IF(XMAX.GT.CF(K,2*M,1))GO TO 31
      IHOLD=1
      GO TO 20
   31 S=1.001
   20 CONTINUE
   21 CONTINUE
      IF(LEVEL)22,23,23
   23 ZA=ZA*45.0/ATAN(1.0)
      WRITE(3,24)ZM,ZA,IRL(K,1,1),IRL(K,2,1),IRL(K,3,1)
   24 FORMAT(///,2E20.6,3I5,5X,'IMPEDANCE AND IRL',///)
   22 RETURN
      END

END OF SEGMENT, LENGTH  1095, NAME  ZRELAY
```

```
      SUBROUTINE RLMOV
      INTEGER FAULT
      DIMENSION Y(3)
      DIMENSION CF(20,8,3),CT(20,4,3),CP(20.3,3),IRL(20,3,3)
      DIMENSION TRIP(20,3),TIME(3),MCLOC(20.2)
      DIMENSION XCOEF(20,3),KOUNT(20)
      COMMON/O/IHOLD
      COMMON/E/NR
      COMMON/D/MCLOC,XCOEF
      COMMON/C1/IRL
      COMMON/C2/CF
      COMMON/C3/CP
      COMMON/C4/CT
      COMMON/C5/TRIPSAVE(20,3)
      COMMON/C6/NOMOVE(20,3)
      COMMON/C7/MCHK
      COMMON/C20/I8
      COMMON/D2/INTERTRIP(20,3)
      COMMON/T4/IPRINT
      EXTERNAL F
      EXTERNAL F1
      EXTERNAL F2
       COMMON/G/Y,TIME,TRIP,KOUNT,XMAX,XMIN-TMAX
      COMMON/A/M
      COMMON NZ(5),FAULT,XC(10),LEVEL
      IHOLD=0
      GO TO(42,7),I8
   42 I8=2
      DT=0.002
      XMIN=0.008
      XMAX=0.01
      DO 20 K=1,NR
      KOUNT(K)=0
      DO 20 I=1,3
      IF(LEVEL-10)41,41,40
   40 CONTINUE
      WRITE(3,100)CT(K,1,I),CT(K,2,I),CT(K,3,I),K,I
      WRITE(3,101)IRL(K,1,I),IRL(K,2,I),IRL(K,3,I),K,I
  101 FORMAT(3I6,5X,2I4,2X,'IRL  K  I')
  100 FORMAT(3F10.4,'  CT',5X,2I5)
   41 CONTINUE
      TRIPSAVE(K,I)=0.0
      NOMOVE(K,I)=0
   20 TRIP(K,I)=0.0
    7 MCHK=-1
   12 ICHK=0
      XMIN=XMIN+DT
      XMAX=XMAX+DT
      MCHK=MCHK+1
      IF(LEVEL.LT.1)GO TO 102
      WRITE(3,16)XMAX
  102 DO 1 M=1,NR
      S=0.0
      IF(KOUNT(M))1,5,1
    5 CONTINUE
      JF=0
      IF(FAULT-2)24,25,25
   24 I=1
```

```
      IF(IRL(M,3,1)-2)26,27,27
   27 CALL ZRELAY(S)
      GO TO 3
   25 I=2
      IF(IRL(M,3,2)-2)26,27,27
   26 CONTINUE
      GO TO(17,18,19),FAULT
   17 CONTINUE
      J1=MOVE(M,I)
      GO TO(32,1),J1
   32 CONTINUE
      CALL SMPSN(F1,S,IER)
      I=1
      GO TO 22
   18 CONTINUE
      J1=MOVE(M,I)
      GO TO(33,1),J1
   33 CONTINUE
      CALL SMPSN(F,S,IER)
      I=2
      GO TO 22
   19 JF=1
      GO TO 18
   22 CONTINUE
      IF(IER)2,3,2
    2 WRITE(3,4)IER
    4 FORMAT(//,5X,18HSIMPSON ERROR CODE  ,I10,/// )
      STOP
    3 CONTINUE
      TRIP(M,I)=TRIP(M,I)+S
      IF(LEVEL-5)28,29,29
   29 CONTINUE
      WRITE(3,105)TRIP(M,I),M,I
  105 FORMAT(F20.4,2I5,5X,'TRIP   M   I')
   28 CONTINUE
      IF(JF)23,1,23
   23 CONTINUE
      J1=MOVE(M,I)
      GO TO(34,1),J1
   34 CONTINUE
      CALL SMPSN(F2,S,IER)
      I=1
      JF=0
      GO TO 22
    1 CONTINUE
    9 DO 11 K=1,NR
      IF(KOUNT(K).NE.0)GO TO 11
      DO 11 I=1,3
      IF(TRIP(K,I).LT.1.0)GO TO 11
      WRITE(3,8)K,I,XMAX
    8 FORMAT(//,5X,'RELAY POSITION',I4,5X,'PHASE',
     2 I4,5X,'HAS A TRIP IN '
     1,'TIME',F10.4,5X,'SECONDS',//)
      KOUNT(K)=1
      ICHK=1
      CALL OPEN(K,6,7)
      CALL OPEN(K,19,20)
      CALL OPEN(K,30,31)
```

```
      IF(TRIP(K,I).EQ.0.0) GO TO 11
      I6=0
      DO 6 J=1,3
      IF(INTERIRIP(K,J).EQ.0)GO TO 6
      I5=INTERTRIP(K,J)
      IF(TRIP(I5,1).GE.1.0)GO TO 6
      TRIP(I5,1)=1.01
      WRITE(3,21)I5
   21 FORMAT(//,5X,'RELAY POSITION',I5,5X,'BREAKER',
     2' OPENED ON INTERTRIP',//)
      IF(I5.GT.K)GO TO 6
      I6=1
    6 CONTINUE
      IF(IPRINT.EQ.0)GO TO 11
      WRITE(3,13)TRIP(K,I),K,I
   13 FORMAT(E20.4,5X,'DISTANCE MOVED BY REIAY',
     1 I4,5X,'PHASE',I4)
   11 CONTINUE
      IF(I6.EQ.1)GO TO 9
      M1=0
      DO 35 K=1,NR
      DO 35 I=1,3
      IF(TRIPSAVE(K,I).EQ.TRIP(K,I))GO TO 36
      NOMOVE(K,I)=1
      M1=1
      GO TO 35
   36 IF(IRL(K,3,I).LT.2)GO  TO 43
      IF(IHOLD.NE.0)M1=1
   43 NOMOVE(K,I)=0
   35 TRIPSAVE(K,I)=TRIP(K,I)
      IF(M1)37,38,37
   38 WRITE(3,39)
   39 FORMAT(///,10X,'NO RELAY MOVEMENIS.   RUN  STOPPED'  )
      STOP
   37 CONTINUE
   16 FORMAT(///,F10.4,5X,' TIME IN SECONDS')
      IF(XMAX-TMAX)10,15,15
   10 IF(ICHK)12,12,14
   14 RETURN
   15 STOP

      RETURN
      END

END OF SEGMENT, LENGTH   920, NAME  RLMOV
```

```
      SUBROUTINE OPEN(K,N1,N2)
      COMMON/D/G(80),B(80)
      COMMON/D/MCLOC(20,2),XCOEF(20,3)
      COMMON/D/XTIME(20,3)
      COMMON/C1/IRL(20,3,3)
      READ(15'13)MCLOC
      READ(15'60)XTIME
      L=0
      I=IRL(K,1,1)
      J=IRL(K,2,1)
    1 READ(N1'I)G
      READ(N2'I)B
      G(I)=G(I)+G(J)
      B(I)=B(I)+B(J)
      G(J)=0.0
      B(J)=0.0
      WRITE(N1'I)G
      WRITE(N2'I)B
      IF(L)2,2,3
    2 L=1
      IT=I
      I=J
      J=IT
      WRITE(3,4)J,I
    4 FORMAT(5X,'LINE',2I3,5X,'OPENED')
      GO TO 1
    3 CONTINUE
      ITT=0
    5 CONTINUE
      DO 6 M=1,20
      IF(MCLOC(M,1)-J)6,7,6
    7 IF(MCLOC(M,2)-I)6,8,6
    6 CONTINUE
      IF(ITT)10,11,10
   11 IT=J
      J=I
      I=IT
      ITT=1
      GO TO 5
    8 READ(15'15)XCOEF
      MCLOC(M,1)=0
      MCLOC(M,2)=0
      DO 12 IM=1,3
      XTIME(M,IM)=0.0
   12 XCOEF(M,IM)=0.0
      I=M
   15 I=I+1
      IF(MCLOC(I,1).EQ.0)GO TO 16
      MCLOC(M,1)=MCLOC(I,1)
      MCLOC(M,2)=MCLOC(I,2)
      DO 14  IM=1,3
      XTIME(M,IM)=XTIME(I,IM)
   14 XCOEF(M,IM)=XCOEF(I,IM)
      IF(I.LT.20)GO TO 15
   16 CONTINUE
      WRITE(3,13)
   13 FORMAT(1H+,29X,'MACHINE O/C')
      WRITE(15'15)XCOEF
      WRITE(15'13)MCLOC
      WRITE(15'60)XTIME
```

```
      WRITE(15'13)MCLOC
      WRITE(15'60)XTIME
   10 RETURN
      END

END OF SEGMENT, LENGTH   451, NAME  OPEN
```

```
      SUBROUTINE CALC(A1,A2,A3)
      COMMON/A/IX
      X=0.01
      IX=0
      J=1
    7 CONTINUE
      DO 1 K=1,10
      R=A1+A2*SQRT(X)+A3*X
      X=X+0.02
      GO TO (3,4),J
    3 CONTINUE
      IF(ABS(R).LT.1.0E-4)R=0.0
      IF(R)6,1,1
    6 J=2
      IX=1
      X=0.0
      GO TO 7
    4 WRITE(3,2)X,R
    1 CONTINUE
    2 FORMAT(20X,2E30.6,5X,'X AND R FROM CALC')
      RETURN
      END

ND OF SEGMENT, LENGTH   140, NAME  CALC
```

```
      SUBROUTINE XPFIT(CP)
      DIMENSION Y(3),X(3),CP(3)
      COMMON/G/Y,X
      COMMON ZX(13),LEVEL
      DY1=Y(2)-Y(1)
      DY2=Y(3)-Y(1)
      DX1=X(2)-X(1)
      DX2=X(3)-X(1)
      DXR1=SQRT(X(2))-SQRT(X(1))
      DXR2=SQRT(X(3))-SQRT(X(1))
      T1=DY2-DX2*DY1/DX1
      T2=DXR2-DXR1*DX2/DX1
      C=T1/T2
      B=(DY1-C*DXR1)/DX1
      A=Y(1)-B*X(1)-C*SQRT(X(1))
      T4=B
      B=C
      C=T4
      T1=A+B*SQRT(X(1))+C*X(1)
      T2=A+B*SQRT(X(2))+C*X(2)
      T3=A+B*SQRT(X(3))+C*X(3)
      IF(LEVEL-4)9,9,10
   10 WRITE(3,8)T1,T2,T3,Y(1),Y(2),Y(3)
    8 FORMAT(3E20.6,5X,'RESULTS',//,3E20.5,
     15X,'DATA FROM XPFIT')
    9 CONTINUE
      CP(1)=A
      CP(2)=B
      CP(3)=C
      CALL CALC(A,B,C)
      RETURN
      END
```

END OF SEGMENT, LENGTH   319, NAME XPFIT

```
      SUBROUTINE XPEQU(I1,I2,I3,I4)
      DIMENSION CP(20,3,3),IRL(20,3,3)
      DIMENSION TIME(3),Y(3),IBUS(2,240)
      DIMENSION CPT(3),C1(240),C2(240),C3(240)
      DIMENSION ANG(20,3,3)
      COMMON/A/IX
      COMMON/C1/IRL
      COMMON/C3/CP
      COMMON/E/NR
      COMMON/D/IBUS,C1,C2,C3
      COMMON/D/ANG
      COMMON/G/Y,TIME
      COMMON XC(13),LEVEL
      PI=4.0*ATAN(1.0)
      READ(37'28)ANG
      READ(37'1)IBUS
      READ(37'I1)C1
      READ(37'I2)C2
      READ(37'I3)C3
      IF(LEVEL-3)9,9,10
   10 CALL DAM(I1,I2,I3)
      DO 11 J=1,2
      DO 11 K=1,10
      WRITE(3,12)IBUS(J,K),J,K
   11 WRITE(3,13)IRL(K,J,I4),K,J,I4
   12 FORMAT(I10,5X,2I4,5X,'IBUS  J  K')
   13 FORMAT(I10,5X,3I4,5X,'IRL  K  J  PHASE')
    9 CONTINUE
      DO 1 K=1,NR
      J=KFIND(IRL(K,1,I4),IRL(K,2,I4))
      IR=J/IABS(J)
      J=IABS(J)
      Y(1)=C1(J)
      Y(2)=C2(J)
      Y(3)=C3(J)
      IF(I1.LT.12)GO TO 20
      CALL CHECKANG(Y,K,1)
      IF(IR)21,20,20
   21 DO 23 M=1,3
      IF(PI.GE.Y(M))GO TO 24
      Y(M)=Y(M)-PI
      GO TO 23
   24 Y(M)=Y(M)+PI
   23 CONTINUE
   20 CALL XPFIT(CPT)
      IF(IX)14,16,14
   14 WRITE(3,17)IRL(K,1,I4),IRL(K,2,I4)
   17 FORMAT(5X,2I4,5X,'THIS LINE HAS NEGATIVE',
     1 ' CURRENT VALUES.  SEE USER MANUAL')
      WRITE(3,25)
   25 FORMAT(5X,'ERROR CODE 9')
   16 CONTINUE
      IF(LEVEL-6)3,3,5
    5 CONTINUE
      WRITE(3,15)Y

   15 FORMAT(3E20.6,5X,'Y FROM EXPEQU')
      WRITE(3,4)CPT,K,I4
```

```
      4 FORMAT(3E20.6,2I4,'COEFFICIENTS',
       1 ' RELAY NUMBER AND PHASE')
      3 CONTINUE
        IF(I1-12)2,6,6
      2 CONTINUE
        DO 8 M=1,3
        CP(K,M,I4)=CPT(M)
      8 CONTINUE
        GO TO 1
      6 CONTINUE
        DO 7 M=1,3
      7 ANG(K,M,I4)=CPT(M)
      1 CONTINUE
        WRITE(37'25)CP
        WRITE(37'28)ANG
        RETURN
        END

END OF SEGMENT, LENGTH   505, NAME  XPEQU
```

```
      FUNCTION THETACHK(A1,A2)
      THETACHK=2
      PI=4.0*ATAN(1.0)
      TA1=A1+2.0*PI
      TA2=A2+2.0*PI
      UL=TA1+PI/6.0
      LL=TA1-PI/2.0
      IF(TA2.GT.UL)RETURN
      IF(TA2.LT.LL)RETURN
      THETACHK=1
      RETURN
      END

END OF SEGMENT, LENGTH    96, NAME  THETACHK
```

```
      FUNCTION THETALIM(BL,UL,A2)
      COMMON ZX(13), LEVEL
      PI2= 8.0*ATAN(1.0)
      C=PI2/360.0
      BBL=BL*C
      UUL=UL*C
      TA2=A2+PI2
      TBL=BBL+PI2
      TUL=UUL+PI2
      IF(TA2.LT.TBL)GO TO 3
      IF(TA2.GT.TUL)GO TO 3
      THETALIM=1
    5 RETURN
    3 THETALIM=2
      IF(LEVEL.GE.8)GO TO 4
      GO TO 5
    4 WRITE(3,6)BL,UL,A2
    6 FORMAT(5X,'TRIP ACTION INHIBITED BY THETALIM',
     1 3F10.4,5X,'LOWER AND UPPER LIMITS. ANGLE')
      GO TO 5
      END

END OF SEGMENT, LENGTH   137, NAME  THETALIM
```

```
      SUBROUTINE VEQU(I1,I2,I3,I4)
      DIMENSION TIME(3),Y(3),CPT(3)
      DIMENSION V1(80),V2(80),V3(80),VCP(20.3,3)
      DIMENSION IRL(20,3,3)
      COMMON/A/IX
      COMMON/C1/IRL
      COMMON/G/Y,TIME
      COMMON/E/NR
      COMMON/D/VCP,V1,V2,V3
      COMMON XC(13),LEVEL
      READ(37'27)IRL
      IF(I1-25)8,9,9
    8 READ(37'21)VCP
      GO TO 10
    9 READ(37'22)VCP
   10 READ(15'I1)V1
      READ(15'I2)V2
      READ(15'I3)V3
      DO 1 K=1,NR
      J=IRL(K,1,I4)
      Y(1)=V1(J)
      Y(2)=V2(J)
      Y(3)=V3(J)
      IF(I1-25)14,15,15
   15 CALL CHECKANG(Y,K,2)
   14 CALL XPFIT(CPT)
      IF(IX)11,12,11
   11 WRITE(3,13)IRL(K,1,I4)
   13 FORMAT(10X,I5,5X,'THIS NODE HAS NEGATIVE VOLTAGE VALUES.'
     1 ' ERROR CODE 10')
   12 CONTINUE
      IF(LEVEL-4)2,3,3
    3 WRITE(3,5)CPT,K,I4
    5 FORMAT(3E20.6,2I4,5X,'COEFFICIENTS FOR VOLTAGE EQUATIONS
     1',/,5X,'RELAY NUMBER AND PHASE')
    2 CONTINUE
      DO 1 M=1,3
    1 VCP(K,M,I4)=CPT(M)
      IF(I1-25)6,7,7
    6 WRITE(37'21)VCP
      RETURN
    7 WRITE(37'22)VCP
      RETURN
      END

END OF SEGMENT, LENGTH   283, NAME  VEQU
```

```
      SUBROUTINE DAM(I1,I2,I3)
      COMMON/D/IC1(2,240),C1(240),C2(240),C3(240)
      READ(37'I1)C1
      READ(37'I2)C2
      READ(37'I3)C3
      DO 1 K=1,10
    1 WRITE(3,2)C1(K),C2(K),C3(K)
    2 FORMAT(3E20.6)
      RETURN
      END

END OF SEGMENT, LENGTH     90, NAME  DAM
```

```
      SUBROUTINE BREAK
      INTEGER BB,TF,FBUS,FAULT
      COMMON N,BB,L,TF,FBUS,FAULT,XX(6),BF,GF,ICT,IEMF,TOL
      COMMON LEVEL
      COMMON/G/Y(3),TIME(3),TRIP(20,3),KOUNT(20)
      COMMON/G/XMAX,XMIN,TMAX
      COMMON/D/MCLOC(20,2),XCOEF(20,3),G(80),B(80)
      COMMON/D/XTIME(20,3)
C       TEST FOR NEARNESS TO TMAX - SAY 1 CYCLE LIMIT.
      DT=(TMAX-XMAX)/2.0
       IF(DT-0.01)1,1,2
    1 WRITE(3,3)XMAX
    3 FORMAT(5X,'THE TIME IS',F10.4,5X,'SECONDS.   THIS'
     2 ,' IS LESS THAN ONE CYCLE TO TMAX',//,
     1 '    THE RUN HAS STOPPED')
      STOP
    2 CONTINUE
      TYME1=XMAX
      TYME2=XMAX+DT
      TYME3=XMAX+DT+DT
      TIME(1)=TYME1
      TIME(2)=TYME2
      TIME(3)=TYME3
      WRITE(3,101)TYME1,TYME2,TYME3
  101 FORMAT(50X,'THE TIME VALUES ARE',3F10.6)
      READ(6'FBUS)G
      READ(7'FBUS)B
      G(FBUS)=G(FBUS)-GF
      B(FBUS)=B(FBUS)-BF
      WRITE(6'FBUS)G
      WRITE(7'FBUS)B
      CALL PSNS2
      CALL ZSSOL
      READ(15'13)MCLOC
      READ(15'60)XTIME
      READ(15'15)XCOEF
      IF(LEVEL.LE.3)GO TO 14
   15 CONTINUE
      WRITE(3,10)((MCLOC(K,J),J=1,2),K=1,20)
   10 FORMAT(2I10,5X,'MCLOC  K  J')
      WRITE(3,11)((XCOEF(K,J),J=1,3),K=1,20)
   11 FORMAT(3E20.7,8X,'XCOEF  K  J')
   30 FORMAT(3E20.7,8X,'XTIME  K  J')
      WRITE(3,30)((XTIME(K,J),J=1,3),K=1,20)
   14 CONTINUE
      T1=XMAX
    7 I=1
      READ(15'13)MCLOC
      READ(15'15)XCOEF
      READ(15'60)XTIME
    5 K=MCLOC(I,1)
      M=MCLOC(I,2)
      A=XCOEF(I,1)
      S=XCOEF(I,2)
      C=XCOEF(I,3)
      IF(T1.LT.XTIME(I,3))GO TO 41
      T1=XTIME(I,3)
   41 CONTINUE
```

```
      X=A+S*SQRT(T1)+C*T1
      IF(LEVEL-11)12,12,16
   16 CONTINUE
      WRITE(3,13)K,M,X
   13 FORMAT(2I10,F10.4,5X,'K   M   X   FROM BREAK')
   12 CONTINUE
      READ(7'K)B
      B(K)=B(K)+B(M)-1.0/X
      B(M)=1.0/X
      WRITE(7'K)B
      READ(7'M)B
      B(M)=B(M)+B(K)-1.0/X
      B(K)=1.0/X
      WRITE(7'M)B
      I=I+1
      IF(I-21)6,4,4
    6 IF(MCLOC(I,1))4,4,5
    4 CONTINUE
      CALL PSA7
      CALL VECTR
      CALL CRNT(1,6,7)
      CALL CRNT(5,19,20)
      CALL CRNT(7,30,31)
      CALL VCOMP(IC)
      CALL ICOMP(IC)
      IC=IC+1
      IF(IC-4)8,9,9
    8 T1=T1+DT
      GO TO 7
    9 CONTINUE
      CALL XPEQU(3,6,9,1)
      CALL XPEQU(4,7,10,2)
      CALL XPEQU(5,8,11,3)
      CALL XPEQU(12,15,18,1)
      CALL XPEQU(13,16,19,2)
      CALL XPEQU(14,17,20,3)
      CALL VEQU(16,19,22,1)
      CALL VEQU(17,20,23,2)
      CALL VEQU(18,21,24,3)
      CALL VEQU(25,28,31,1)
      CALL VEQU(26,29,32,2)
      CALL VEQU(27,30,33,3)
      RETURN
      END

END OF SEGMENT, LENGTH   778, NAME  BREAK
```

```
      SUBROUTINE SMPSN(F,S,IER)
      COMMON/G/Y(3),TIME(3),TRIP(20,3),KOUNT(20),XMAX,XMIN
      IMAX=100
      DEL=0.01
      A=XMIN
      B=XMAX
      SI1=0.0
      S=0.0
      N=0
      BA=B-A
      IF(BA)20,19,20
   19 IER=1
      RETURN
   20 CONTINUE
      IF(DEL)22,22,23
   22 IER=2
      RETURN
   23 IF(IMAX-1)24,24,25
   24 IER=3
      RETURN
   25 X=BA/2.0+A
      NHALF=1
       IF(F(A)-1.0E-6)1,1,2
    2 CONTINUE
      SUMK=F(X)*BA*2.0/3.0
      S=SUMK+(F(A)+F(B))*BA/6.0
      DO 28 I=2,IMAX
      SI1=S
      S=(S-SUMK/2.0)/2.0
      NHALF=NHALF*2
      ANHLF=NHALF
      FRSTX=A+(BA/ANHLF)/2.0
      SUMK=F(FRSTX)
      XK=FRSTX
      KLAST=NHALF-1
      FINC=BA/ANHLF
      DO 26 K=1,KLAST
      XK=XK+FINC
   26 SUMK=SUMK+F(XK)
      SUMK=SUMK*2.0*BA/(3.0*ANHLF)
      S=S+SUMK
   27 IF(ABS(S-SI1)-ABS(DEL*S))29,28,28
   28 CONTINUE
      IER=4
      GO TO 30
   29 IER=0
   30 N=2*NHALF
    1 CONTINUE
  100 FORMAT(E50.6,2X,1HS)
      RETURN
      END

END OF SEGMENT, LENGTH   294, NAME  SMPSN
```

```
FORTRAN COMPILATION BY #XFAT MK 4C     DATE  13/05/75  TIME  17/14/46


0001                 READ FROM(ED,FILE3(18).FORTPSAWH)
SUBFILE FORTPSAWH
0001                 SEND TO(PSAB)
0002                 OVERLAY SEGMENTS
0003                 TRACE 2
0004                 COMPRESS INTEGER AND LOGICAL
0005                 END
```

```
      SUBROUTINE PSA2
      INTEGER BB,C,D,TF
      INTEGER BRI1(240),BRI2(240)
      DIMENSION BRR1(240),BRR2(240)
      COMMON N,BB,L,TF
      COMMON/D/BRI1,BRI2,BRR1,BRR2
      COMMON/T2/CARD(10)
      COMMON/T3/K3
      CALL DEFBUF(29,80,CARD)
      WRITE(3,1)
    1 FORMAT(1H1,////,6X,'BRANCH DATA',//,6X,
     1 'BRANCH',5X,'RESISTANCE',5X,'REACTANCE')
      DO 116 M=1,BB
      CALL FREAD
      READ(29,117)C,D,R,X
      WRITE(3,3)C,D,R,X
    3 FORMAT(I8,I4,5X,F10.4,5X,F9.4)
  117 FORMAT(2I4,2F10.4)
      BRI1(M)=C
      BRI2(M)=D
      BRR1(M)=R/(R**2+X**2)
      P1=R**2
      P2=X**2
      P3=P1+P2
      P4=-X/P3
      BRR2(M)=P4
  116 CONTINUE
      WRITE(1'1)BRI1
      WRITE(1'2)BRI2
      WRITE(1'3)BRR1
      WRITE(1'5)BRR2
      L=0
      RETURN
      END
```

END OF SEGMENT, LENGTH   186, NAME PSA2

```
      SUBROUTINE PSA3
      INTEGER T,W,BB,TF
      INTEGER SORT(240),BRI1(240)
      COMMON N,BB,L,TF
      COMMON/D/SORT,BRI1
      DO 1008 I=1,BB
 1008 SORT(I)=I
      IF(L)1030,1030,1031
 1031 READ(1'2)BRI1
      GO TO 1032
 1030 READ(1'1)BRI1
 1032 CONTINUE
      DO 1006 I=1,BB
      K=BB-I
      DO 1006 J=1,K
      T=SORT(J)
      W=SORT(J+1)
      IF(BRI1(T)-BRI1(W))1006,1006,1007
 1007 SORT(J)=W
      SORT(J+1)=T
 1006 CONTINUE
      WRITE(1'7)SORT
      RETURN
      END

END OF SEGMENT, LENGTH   160, NAME  PSA3
```

```
      SUBROUTINE PSA5
      INTEGER S,BB ,TF
      DIMENSION G(80),B(80),YL(80),ANGLE(80)
      COMMON N,BB,L,TF
      COMMON/D/G,B,YL,ANGLE
      READ(6'1)G
      READ(7'1)B
      READ(10'5)YL
      READ(10'4)ANGLE
      G(1)=YL(1)*COS(ANGLE(1))
      B(1)=YL(1)*SIN(ANGLE(1))
      DO 118 K=2,N
      G(1)=G(1)-G(K)
  118 B(1)=B(1)-B(K)
      WRITE(6'1)G
      WRITE(7'1)B
      DO 119 K=2,N
      READ(6'K)G
      READ(7'K)B
      G(K)=YL(K)*COS(ANGLE(K))
      B(K)=YL(K)*SIN(ANGLE(K))
      S=K-1
      DO 120 M=1,S
      G(K)=G(K)-G(M)
  120 B(K)=B(K)-B(M)
      IF(K-N)97,121,97
   97 S=K+1
      DO 121 M=S,N
      G(K)=G(K)-G(M)
      B(K)=B(K)-B(M)
  121 CONTINUE
      WRITE(6'K)G
      WRITE(7'K)B
  119 CONTINUE
      RETURN
      END

END OF SEGMENT, LENGTH   386, NAME  PSA5
```

```
      SUBROUTINE PSA6
      INTEGER BB,TF,FBUS,FAULT,      D
       DIMENSION  G(80),B(80)
      DIMENSION IBUS(2,240)
       COMMON N,BB,L,TF,FBUS,FAULT,RN,XN,RZ.XZ,R,X,BF,GF
      COMMON ICT
      COMMON/D/G,B,IBUS
      COMMON/T2/CARD(10)
      COMMON/T3/K3
      CALL DEFBUF(29,80,CARD)
      DO 20 K=1,BB
       IBUS(1,K)=0
   20 IBUS(2,K)=0
      IF(TF)5,5,2
    2 WRITE(3,4)
    4 FORMAT(////,7X,'T/F   K   M',//)
      DO 1 D=1,TF
      CALL FREAD
      READ(29,3)T1,K,M
      WRITE(3,3)T1,K,M
    3 FORMAT(F10.4,2I3)
      READ(6'K)G
      READ(7'K)B
      G(K)=G(K)+G(M)-T1*T1*G(M)
      B(K)=B(K)+B(M)-T1*T1*B(M)
      G(M)=G(M)*T1
      B(M)=B(M)*T1
      WRITE(6'K)G
      WRITE(7'K)B
      T2=G(M)
      READ(6'M)G
      G(K)=T2
      T2=B(M)
      READ(7'M)B
      B(K)=T2
      WRITE(6'M)G
      WRITE(7'M)B
    1 CONTINUE
    5 CONTINUE
      I=1
      J=N-1
      DO 11 K=1,J
      READ(6'K)G
      READ(7'K)B
      I1=K+1
      DO 11 M=I1,N
      IF(G(M))8,9,8
    9 IF(B(M))8, 11,8
    8 IBUS(1,I)=K
      IBUS(2,I)=M
      I=I+1
   11 CONTINUE
      WRITE(37'1)IBUS
      RETURN
      END

END OF SEGMENT, LENGTH   405, NAME  PSA6
```

```
      SUBROUTINE PSA4
      INTEGER S,C,D,T,BB,TF
      INTEGER BRI1(240),BRI2(240),SORT(240)
      DIMENSION BRR1(240),BRR2(240),G(80),B(80)
      COMMON N,BB,L,TF
      COMMON/D/SORT,BRI1,BRI2,BRR1,BRR2,G,B
      S=0
      T=0
      READ(1'7)SORT
      READ(1'3)BRR1
      READ(1'5)BRR2
      IF(L)1033,1033,1034
 1033 READ(1'1)BRI1
      READ(1'2)BRI2
      GO TO 1035
1034  CONTINUE
      READ(1'2)BRI1
      READ(1'1)BRI2
 1035 CONTINUE
      DO 1001 I=1,BB
      IF(I-BB)1015,1016,1016
 1016 D=SORT(BB)
      S=1
      GO TO 1014
 1015 C=SORT(I+1)
      D=SORT(I)
      IF(BRI1(C)-BRI1(D))1014,1014,1013
 1013 S=1
 1014 CONTINUE
      IF(T)1040,1040,1041
 1040 K=BRI1(D)
      READ(6'K)G
      READ(7'K)B
      DO 3 J=1,N
    3 CONTINUE
 1041 CONTINUE

      J=BRI2(D)
      G(J)=-BRR1(D)
      B(J)=-BRR2(D)
      IF(S)8,8,1003
 1003 CONTINUE
      WRITE(6'K)G
      WRITE(7'K)B
      S=0
      T=0
      GO TO 1001
    8 T=1
 1001 CONTINUE
      L=2
      RETURN
      END

END OF SEGMENT, LENGTH   308, NAME  PSA4
```

```
      SUBROUTINE PSA8
      INTEGER BB,TF,1
      REAL ILMOD,IBASE
      REAL IP,IQ,IMOD,MW,MVAR,MW1,MVAR1 ,IMOD
      DIMENSION VP(80),VQ(80),ANGLE(80),YL(80),PS(80),QS(80)
      DIMENSION G(80),B(80)
      COMMON N,BB,L,TF
      COMMON/D/VP,VQ,ANGLE,YL,PS,QS,G,B
      COMMON/C10/ISCIND,NCOUNT
      COMMON/C11/NODESC(20)
      VBASE=1.0
      IBASE=1.0
      YNC=0.0
       WRITE(3,333)
  333 FORMAT(////,7X,'NET GENERATIONS',
     11X,'OR LOAD AT BUSBARS'//16X,'P.U.VOLTAGES',
     219X,'P.U. GENERATION',21X,'P.U. LOAD'.//
     35X,'BUS',8X,'MOD',4X,'ANGLE (DEG)',11X,'MW',
     412X,'MVAR',16X,'MW',12X,'MVAR')
      READ(10'1)PS
      READ(10'2)QS
      READ(15'1)VP
      READ(15'2)VQ
      READ(10'4)ANGLE
      READ(10'5)YL
      DO 144 K=1,N
      CALL LMT(VP(K),VQ(K),VMOD,R)
      R=57.29578*R
      VMOD=VMOD+YNC
      THETA=R+ANGLE(K)
      ILMOD=VMOD*YL(K)
      IF(PS(K))163,163,128
  163 PS(K)=PS(K)*VBASE*IBASE +YNC
      QS(K)=QS(K)*VBASE*IBASE +YNC
      PS(K)=-1.0*PS(K)
      QS(K)=-1.0*QS(K)
      WRITE(3,335)K,VMOD,R,PS(K),QS(K)
  335 FORMAT(I8,F13.4,F10.2,44X,F9.4,F14.4)
      GO TO 144
  128 PS(K)=PS(K)*VBASE*IBASE +YNC
      QS(K)=QS(K)*VBASE*IBASE+YNC
      WRITE(3,336)K,VMOD,P,PS(K),QS(K)
  336 FORMAT(I8,F13.4,F10.2,F19.4,F14.4)
  144 CONTINUE
      WRITE(3,5)
    5 FORMAT(1H1)
      WRITE(3,337)
  337 FORMAT(////,5X,'LINF FLOWS.   (ALL VALUES IN P.U. )',
     2 ///,
     1 5X,'LINE',25X,'SEND',32X,'RECEIVE',27X,'CURRENT',
     3 //,25X,'MW',13X,'MVAR',19X,'MW',14X,'MVAR',
     4 15X,'MOD',9X,'ANGLE')
      J=N-1
      DO 129 M=1,J
      READ(6'M)G
      READ(7'M)B
      K=M+1
      DO 129 I=K,N
```

```
      IF(G(T))502,503,502
  503 IF(B(T))502,129,502
  502 T11=VP(M)-VP(T)
      T12=VQ(M)-VQ(T)
      IP=(T11*(-G(T))-T12*(-B(T)))*IBASE
      IQ= (T12*(-G(T))+T11*(-B(T)))*IBASE
      MW=(IP*VP(M)+IQ*VQ(M)) +YNC
      MVAR=(IP*VQ(M)-IQ*VP(M)) +YNC
      MW1=(IP*VP(T)+IQ*VQ(T))   +YNC
      MVAR1=(IP*VQ(T)-IQ*VP(T)) +YNC
      CALL LMT(IP,IQ,IMOD,THETA)
      THETA=THETA*57.29578
      WRITE(3,338)M,T,MW,MVAR,MW1,MVAR1,IMOD,THETA
  338 FORMAT(I6,' - ',I2,5X,2F15.4,7X,2F16.4,10X,F9.4,F12.2)
  129 CONTINUE
      IF(ISCIND)1,1,2
    2 IF(NCOUNT.LT.1)RETURN
      M2=NODESC(NCOUNT)
      READ(6'M2)G
      READ(7'M2)B
      T1=0.0
      T2=0.0
      DO 3 K=1,N
      T1=T1+G(K)*VP(K)-B(K)*VQ(K)
    3 T2=T2+B(K)*VP(K)+G(K)*VQ(K)
      T1 =SQRT(T1*T1+T2*T2)
      WRITE(3,4)M2,T1
    4 FORMAT(///,5X,'THREE PHASE-SHORT CIRCUIT AT',
     2 'BUS',I4,//,5X,
     1   ' TOTAL 3-PHASE SCMVA  (P.U.)  =',F10.4)
    1 CONTINUE
      RETURN
      END

ND OF SEGMENT, LENGTH   701, NAME  PSA8
```

```
      SUBROUTINE PSA7
      INTEGER Z    ,BB,TF ,FBUS,FAULT
      REAL IP,IO,MODV(80)
      DIMENSION PS(80),QS(80),VP(80),VQ(80)-G(80),B(80)
      DIMENSION ND7(80),MPS(3,80)
      DIMENSION  Y(40,41)
       COMMON N,BB,L,TF,FBUS,FAULT,RN,XN,RZ-XZ,R,X,BF,GF ,ICT
      COMMON IEMF,TOL,LEVEL
      COMMON/D/VP,VQ,VP1(80),VQ1(80),VP2(80),VQ2(80)
      COMMON/D/MODV,PS,QS,G,B,ND7,MPS,Y
      COMMON/C10/ISCIND,NCOUNT,ND7IEMP
      COMMON/C11/NODESC(20)
      IX=0
      IB=20
 2000 FORMAT(5X,'SUBROUTINE PSA7 IN OPERATION')
      IF(LEVEL-3)42,43,43
   43 CONTINUE
       WRITE(3,2000)
   10 FORMAT(I6,2F15.5,I10)
      WRITE(3,1)ICT,IEMF,LEVEL,TOL
    1 FORMAT(3I10,E20.6,5X,'ICT IEMF LEVEL  TOL')
   42 CONTINUE
      II=50
      CALL NAM(6,7)
      READ(10'3)MODV
      READ(10'1)PS
      READ(10'2)QS
      READ(23'2)ND7
      IMAX=600
      I=0
   33 FORMAT(I3)
      READ(15'3)VP
      READ(15'4)VQ
      DO 4 K=1,N
      READ(6'K)G
      DO 4 J=K,N
    4 Y(K,J)=G(J)
      DO 14 K=2,N
      READ(7'K)B
      Y(N+1,K)=B(K)
      KT=K-1
      DO 14 J=1,KT
   14 Y(K,J)=B(J)
      READ(7'1)B
      Y(N+1,1)=B(1)
      DO 2 K=1,N
      IT=1
      READ(6'K)G
      READ(7'K)B
      DO 3 J=1,N
      IF(K-J)20,3,20
   20 IF((ABS(G(J))+ABS(B(J)))-0.0001)9,3,3
    9 IT=IT+1
    3 CONTINUE
      IF(IT-N)2,16,2
   16 ND7(K)=0
      VP(K)=0.0
      VQ(K)=0.0
```

```
      WRITE(3,1112)K
 1112 FORMAT(////,5X,'NODE',I4,' IS AN ISOLATED NODE',
     2 ' VOLTAGE AND ND7 SET TO ZERO',///)
    2 CONTINUE
      IF(ISCINO.EQ.0)GO TO 1111
      NCOUNT=NCOUNT+1
      IF(NCOUNT.EQ.0)GO TO 1111
      IF(NCOUNT.EQ.1)GO TO1311
      IF(NCOUNT.EQ.21)STOP
      IF(NODESC(NCOUNT).EQ.0)STOP
      J=NODESC(NCOUNT-1)
      ND7(J)=ND7TEMP
 1311 J=NODESC(NCOUNT)
      DO 8 M=1,N
      IF(ND7(M).LT.1)GO TO 8
      VP(M)=0.5
      VQ(M)=-0.05
    8 CONTINUE
      VP(J)=0.0
      VQ(J)=0.0
      ND7TEMP=ND7(J)
      ND7(J)=0
 1111 CONTINUE
      KK=0
   11 Z=0
      K=0
  122 K=K+1
      IF(ND7(K))40,41,40
   41 Z=Z+1
       GO TO 61
   40 CONTINUE
       IP=0.
      1Q=0.
      IF(K-1)201,202,201
  201 KT=K-1
      DO 204 M=1,KT
      IP=(IP+Y(M,K)*VP(M)-Y(K,M)*VQ(M))
      IQ=(IQ+Y(K,M)*VP(M)+Y(M,K)*VQ(M))
  204 CONTINUE
      IF(K-N)202,205,205
  202 KT=K+1
      DO 205 M=KT,N
      IP=(IP+Y(K,M)*VP(M)-Y(M,K)*VQ(M))
      IQ=(IQ+Y(M,K)*VP(M)+Y(K,M)*VQ(M))
  205 CONTINUE
      IP=IP+Y(K,K)*VP(K)-Y(N+1,K)*VQ(K)
      IQ=IQ+Y(N+1,K)*VP(K)+Y(K,K)*VQ(K)
      P=IP*VP(K)+IQ*VQ(K)
      Q=IP*VQ(K)-IQ*VP(K)
      IF(ND7(K))166,168,160
  168 PS(K)=P
  166 QS(K)=Q
  160 IF(KK)170,170,124
  170 CONTINUE
      DP=PS(K)-P
      DQ=QS(K)-Q
      T1=VQ(K)*Y(K,K)
      T2=VQ(K)*Y(N+1,K)
```

```
      T3=VP(K)*Y(K,K)
      T4=VP(K)*Y(N+1,K)
      T5=T1-T4
      T6=T3+IP
      A=T3+IP+T2
      H=IQ-T4+T1
      E=T1-T4-IQ
      F=IP-T3-T2
      IF(ND7(K))127,126,126
  127 T81=T5+IQ
      T82=T6+T2
      T83=VP(K)**2+VQ(K)**2
      T84=(MODV(K)-T83)/2.
      T85=T81*VP(K)-T82*VQ(K)
      DVP=(T81*T84-VQ(K)*DP)/T85
      DVQ=(-1)*(T82*T84-VP(K)*DP)/T85
      GO TO 570
  126 T7=T6*T6-T2*T2+IQ*IQ-T5*T5
      T8=A*F-H*E
      DVP=(F*DP-H*DQ)/T8
      DVQ=(A*DQ-E*DP)/T8
  570 VP(K)=VP(K)+DVP
      VQ(K)=VQ(K)+DVQ
  571 CONTINUE
      IF(ABS(DVP)-TOL    )60,60,61
   60 IF(ABS(DVQ)-TOL)62,62,61
   62 Z=Z+1
   61 CONTINUE
      IF(K-N)122,161,122
  161 I=I+1
      CALL ACCL1(I8,IX,I)
      IF(I.LT.II)GO TO 36
      II=II+50
      DO 37 K=1,N
      IF((ABS(VP(K))+ABS(VQ(K))).GT.1.0E-6)GO TO 37
      IF(ISCIND.NE.0)GO TO 37
      WRITE(3,1113)K,ND7(K)
 1113 FORMAT(//,5X,'VOLTAGE AT NODE',I4,5X,'IS ZERO.',
     1 'ND7 FOR THIS NODE IS',I6,///)
   37 WRITE(3,1001)K,VP(K),VQ(K),I
 1001 FORMAT(I10,2F20.9,I10)
      WRITE(3,1002)
 1002 FORMAT(//)
   36 CONTINUE
      IF(I-IMAX)702,701,701
  701 CONTINUE
      WRITE(3,6)I
    6 FORMAT(I12,2X,'ITERATIONS',2X,'THESE VALUES',
     1 ' ARE NOT A SOLUTION')
      DO 5 J=1,N
      WRITE(3,7)J,VP(J),VQ(J)
    7 FORMAT(I4,2F12.5)
    5 CONTINUE
      CALL EXIT
  702 CONTINUE
      IF(Z-N)11,130,130
  130 WRITE(3,333) I
  333 FORMAT(1H1,I6,5X,'ITERATIONS')
```

```
      KK=1
       K=1
      GO TO 40
  124 CONTINUE
      DO 171 K=1,N
      IF(ND7(K))171,172,171
  172 CONTINUE
      IP=0.
      IQ=0.
      IF(K-1)401,402,401
  401 KT=K-1
      DO 404 M=1,KT
      IP=(IP+Y(M,K)*VP(M)-Y(K,M)*VQ(M))
      IQ=(IQ+Y(K,M)*VP(M)+Y(M,K)*VQ(M))
  404 CONTINUE
      IF(K-N)402,405,405
  402 KT=K+1
      DO 405 M=KT,N
      IP=(IP+Y(K,M)*VP(M)-Y(M,K)*VQ(M))
      IQ=(IQ+Y(M,K)*VP(M)+Y(K,M)*VQ(M))
  405 CONTINUE
      IP=(IP+Y(K,K)*VP(K)-Y(N+1,K   )*VQ(K))
      IQ=IQ+Y(N+1,K)*VP(K)+Y(K,K)*VQ(K)
      Q=IP*VQ(K)-IQ*VP(K)
      P=IP*VP(K)+IQ*VQ(K)
      PS(K)=P
      QS(K)=Q
  171 CONTINUE
      WRITE(10'1)PS
      WRITE(10'2)QS
      READ(23'4)MPS
      CALL ROTAT(VP,VQ,MPS,1)
      WRITE(15'1)VP
      WRITE(15'2)VQ
      IF(IEMF)34,35,34
   34 CALL EMF(IEMF)
   35 CONTINUE
      DO 13 K=1,N
      IF(ND7(K))12,13,12
   12 ND7(K)=1
   13 CONTINUE
      WRITE(23'2)ND7
      RETURN
      END

END OF SEGMENT, LENGTH  1881, NAME  PSA7
```

```
      SUBROUTINE PSNS2
       INTEGER TF
      INTEGER BB,FBUS,FAULT,ND(80) ,Z
      REAL IP,IO     ,IPN,IQN
      DIMENSION G(80),B(80),VP(80),VQ(80)
      DIMENSION  Y(40,41)
       COMMON N,BB,L,TF,FBUS,FAULT,RN,XN,RZ.XZ,R,X,BF,GF
      COMMON ICT,IEMF,TOL,LEVEL
      COMMON/D/VP,VQ,VP1(80),VQ1(80),VP2(80),VQ2(80),ND,G,B,Y
      IX=0
      IB=8
      CALL NAM(19,20)
      II=50
      IMAX=600
      I=0
       READ(23'1)ND
      WRITE(3,8)FBUS,FAULT
    8 FORMAT(    2I6,  ' FBUS     FAULT  ',///)
      ND(FBUS)=0
      DO 2 M=1,N
      IF(ND(M))3,44,44
    3 VP(M)=0.0
      GO TO 2
   44 VP(M)=0.6
    2 VQ(M)=0.0
      VP(FBUS)=1.0
      DO 4 K=1,N
      READ(19'K)G
      DO 4 J=K,N
    4 Y(K,J)=G(J)
      DO 14 K=2,N
      READ(20'K)B
      Y(N+1,K)=B(K)
      KT=K-1
      DO 14 J=1,KT
   14 Y(K,J)=B(J)
      READ(20'1)B
      Y(N+1,1)=B(1)
      IF(ABS(Y(FBUS,FBUS)).GT.1.0E-4)GO TO 11
      IF(ABS(Y(N+1,FBUS)).GT.1.0E-4)GO TO 11
      IF(Y(FBUS,FBUS).GT.1.0E-4)GO TO 11
      IF(Y(N+1,FBUS).GT.1.0E-4)GO TO 11
      WRITE(3,899)Y(FBUS,FBUS),Y(N+1,FBUS)
  899 FORMAT(2E30.6)
      GO TO 7
   11  Z=1
      IF(I-IMAX)16,17,18
   18 CALL EXIT
   17 DO 19 M=1,N
      WRITE(3,20)VP(M),VQ(M),M
   20 FORMAT(2F12.5,I10)
   19 CONTINUE
   16 CONTINUE
       DO 122 K=1,N
       IF(ND(K))9,122,21
    9 Z=Z+1
      GO TO 122
   21 CONTINUE
```

```
      IP=0.
      IQ=0.
      IF(K-1)201,202,201
  201 KT=K-1
      DO 204 M=1,KT
      IP=(IP+Y(M,K)*VP(M)-Y(K,M)*VQ(M))
      IQ=(IQ+Y(K,M)*VP(M)+Y(M,K)*VQ(M))
  204 CONTINUE
      IF(K-N)202,205,205
  202 KT=K+1
      DO 205 M=KT,N
      IP=(IP+Y(K,M)*VP(M)-Y(M,K)*VQ(M))
      IQ=(IQ+Y(M,K)*VP(M)+Y(K,M)*VQ(M))
  205 CONTINUE
      IP=IP+Y(K,K)*VP(K)-Y(N+1,K)*VQ(K)
      IQ=IQ+Y(N+1,K)*VP(K)+Y(K,K)*VQ(K)
      P=IP*VP(K)+IQ*VQ(K)
      Q=IP*VQ(K)-IQ*VP(K)
      DP=-P
      DQ=-Q
      Y1=Y(K,K)*VP(K)
      Y2=Y(N+1,K)*VP(K)
      Y3=Y(K,K)*VQ(K)
      Y4=Y(N+1,K)*VQ(K)
      A=IP+Y1+Y4
      AB=IQ-Y2+Y3
      C=Y3-Y2-IQ
      D=IP-Y1-Y4
      IF(ABS(A*D-AB*C)-1.0E-5)5,5,1
    5 VP(K)=0.0
      VQ(K)=0.0
      GO TO 68
    1 CONTINUE
      DVP=(D*DP-AB*DQ)/(A*D -AB*C)
      DVQ=(-C*DP+A*DQ)/(A*D-AB*C)
      IF(ABS(DVP)-TOL)66,66,67
   66 IF(ABS(DVQ)-TOL)68,68,67
   67 CONTINUE
      VP(K)=VP(K)+DVP
      VQ(K)=VQ(K)+DVQ
      GO TO 122
   68 Z=Z+1
  122 CONTINUE
      CALL ACCL1(IB,IX,I)
      I=I+1
      IF(I.LT.II)GO TO 36
      II=II+50
      DO 37 K=1,N
   37 WRITE(3,10)K,VP(K),VQ(K),I
   10 FORMAT(20X,I10,2E20.6, 5X,11HK   VPN   VQN   )
   36 CONTINUE
      IF(Z-N)11,12,12
   12 CONTINUE
      WRITE(3,13)I
   13 FORMAT(I4,6X,'ITERATIONS FOR NEGATIVE NETWORK' ///
     C 5X, ' THE NEGATIVE SEQUENCE IMPEDANCE IS')
      K=FBUS
      IPN=0.
```

```
      IQN=0.
      IF(FBUS-1)301,302,301
  301 KT=FBUS-1
      DO 304 M=1,KT
      IPN=(IPN+Y(M,K)*VP(M)-Y(K,M)*VQ(M))
      IQN=(IQN+Y(K,M)*VP(M)+Y(M,K)*VQ(M))
  304 CONTINUE
      IF(K-N)302,305,305
  302 KT=FBUS+1
      DO 305 M=KT,N
      IPN=(IPN+Y(K,M)*VP(M)-Y(M,K)*VQ(M))
      IQN=(IQN+Y(M,K)*VP(M)+Y(K,M)*VQ(M))
  305 CONTINUE
      IPN=IPN+Y(FBUS,FBUS)*VP(FBUS)-Y(N+1,FBUS)*VQ(FBUS)
      IQN=IQN+Y(N+1,FBUS)*VP(FBUS)+Y(FBUS,FBUS)*VQ(FBUS)
      RN=XMOD(IPN,IQN)
      XN=-XMOD(IQN,IPN)
   15 FORMAT (1H+,50X,F8.4,2X,'+',2X,'J(',2X,F8.4,1X,')')
      IF(ABS(RN)-1000.0)6,7,7
    6 IF(ABS(XN)-1000.0)22,7,7
    7 WRITE(3,23)
   23 FORMAT(///,5X,'FAULT ISOLATED.  RUN STOPPED',///)
      STOP
   22 CONTINUE
      WRITE(3,15)RN,XN
      WRITE(15'9)VP
      WRITE(15'10)VQ
      IF(LEVEL-1)432,433,433
  433 CONTINUE
      WRITE(3,246)
  246 FORMAT(1H1)
      DO 110 M=1,N
      WRITE(3,111)VP(M),VQ(M),M
  111 FORMAT(///2E20.5,I8)
  110 CONTINUE
  432 CONTINUE
      RETURN
      END
```

END OF SEGMENT, LENGTH 1439, NAME PSNS2

```
      SUBROUTINE PSAZS
      INTEGER TF
      INTEGER BB,FBUS,FAULT,NDZ(80),TFCON
      DIMENSION G(80),B(80)
      COMMON N,BB,L,TF,FBUS,FAULT,RN,XN,RZ.XZ,R,X,BF,GF
      COMMON ICT,IEMF,TOL,LEVEL
      COMMON/D/NDZ,G,B
      COMMON/T2/CARD(10)
      COMMON/T3/K3
      READ(23'3)NDZ
      CALL DEFBUF(29,80,CARD)
      IF(LEVEL-2)10,13,13
   13 CONTINUE
      WRITE(3,2500)R,RN,RZ,X,XN,XZ
 2500 FORMAT(6E17.6)
   10 CONTINUE
      CALL FREAD
      READ(29,31)ICCZS
   31 FORMAT(I3)
      IF(ICCZS)1,1,2
    2 CONTINUE
      DO 14 I=1,ICCZS
      J=0
      CALL FREAD
      READ(29,3)K,M,TFCON,ZR,ZX,XZS,XZT,MN,NT
    3 FORMAT(3I3,4F8.4,2I3)
      WRITE(3,32)K,M,TFCON,ZR,ZX,XZS,XZT,MN.NT
   32 FORMAT('  K M TFCON   ZR   ZX   XZS   XZT   MN   NT   ',
     1 3I4,4F10.4,2I5,//)
      IF(TFCON.LE.3)GO TO 105
      ITFC=TFCON-3
      GO TO(101,102,103,104),ITFC
  105 CONTINUE
      GZ=XMOD(ZR,ZX)
      BZ=-XMOD(ZX,ZR)
    4 CONTINUE
      READ(30'K)G
      READ(31'K)B
      GO TO (7,7,19),TFCON
    7 G(K)=G(K)+G(M)+GZ
      B(K)=B(K)+B(M)+BZ
      IF(TFCON-2)6,30,30
   19 G(K)=G(K)+GZ
      B(K)=B(K)+BZ
      J=1
      NDZ(K)=1
      GO TO 11
   30 G(M)=-GZ
      B(M)=-BZ
      GO TO 11
    6 G(M)=0.0
      B(M)=0.0
      GZ=0.0
      BZ=0.0
   11 WRITE(30'K)G
      WRITE(31'K)B
      IF(J)12,12,14
   12 KTEMP=K
```

```
      K=M
      M=KTEMP
      J=1
      GO TO 4
      GO TO 14
  101 CONTINUE
      READ(31'K)B
      B(K)=B(K)+B(MN)
      B(MN)=0.0
      WRITE(31'K)B
      READ(31'MN)B
      B(MN)=B(MN)+B(K)+B(NT)+B(M)
      B(K)=0.
      B(NT)=0.
      B(M)=0.
      WRITE(31'MN)B
      READ(31'M)B
      B(M)=B(M)+B(MN)
      B(MN)=0.
      WRITE(31'M)B
      READ(31'NT)B
      B(NT)=B(NT)+B(MN)
      B(MN)=0.
      WRITE(31'NT)B
      GO TO 14
  102 READ(31'K)B
      B(K)=B(K)+B(MN)
      B(MN)=0.0
      WRITE(31'K)B
      READ(31'MN)B
      B(MN)=B(MN)+B(K)+B(NT)+B(M)-1.0/XZS-1.0/XZT
      B(K)=0.
      B(M)=1.0/XZS
      B(NT)=0.
      WRITE(31'MN)B
      READ(31'NT)B
      B(NT)=B(NT)+B(MN)
      B(MN)=0.
      WRITE(31'NT)B
      READ(31'M)B
      B(M)=B(M)+B(MN)-1.0/XZS
      B(MN)=1.0/XZS
      WRITE(31'M)B
      GO TO 14
  103 READ(31'K)B
      B(K)=B(K)+B(MN)-1.0/ZX
      B(MN)=1.0/ZX
      WRITE(31'K)B
      READ(31'MN)B
      B(MN)=B(MN)+B(K)+B(NT)+B(M)-1.0/ZX-1.0/XZT-1.0/XZS
      B(NT)=1.0/XZT
      B(K)=1.0/ZX
      B(M)=1.0/XZS
      WRITE(31'MN)B
      READ(31'NT)B
      B(NT)=B(NT)+B(MN)
      B(MN)=0.0
      WRITE(31'NT)B
```

```
      READ(31'M)B
      B(M)=B(M)+B(MN)-1.0/XZS
      B(MN)=1.0/XZS
      WRITE(31'M)B
      GO TO 14
  104 READ(20'K)B
      B(K)=B(K)+B(MN)
      B(MN)=0.
      WRITE(31'K)B
      READ(31'MN)B
      B(MN)=B(MN)+B(K)+B(NT)+B(M)-1.0/XZS-1.0/XZT-1.0/ZX
      B(K)=0.
      B(NT)=0.
      B(M)=1.0/XZS
      WRITE(31'MN)B
      READ(31'NT)B
      B(NT)=B(NT)+B(MN)
      B(MN)=0.
      WRITE(31'NT)B
      READ(31'M)B
      B(M)=B(M)+B(MN)-1./XZS
      B(MN)=1.0/XZS
      WRITE(31'M)B
   14 CONTINUE
    1 CONTINUE
      NDZ(FBUS)=0
      DO 100 K=1,N
        READ(30'K)G
       READ(31'K)B
      BAT=(ABS(G(K))+ABS(B(K)))
      IF(BAT-0.0001)8,8,9
    8 NDZ(K)=-1
    9 CONTINUE
  100 CONTINUE
      WRITE(23'3)NDZ
      RETURN
      END

END OF SEGMENT, LENGTH  1397, NAME  PSAZS
```

```
      SUBROUTINE ZSSOL
      INTEGER BB,TF,Z,FAULT,FBUS,NDZ(80)
      REAL IP,IQ,IPZ,IQZ
      DIMENSION G(80),B(80),VP(80),VQ(80)
      DIMENSION  Y(40,41)
      DIMENSION NDZT(80),IA2(80)
       COMMON N,BB,L,TF,FBUS,FAULT,RN,XN,RZ.XZ,R,X,BF,GF
      COMMON ICT,IEMF,TOL,LEVEL
      COMMON/D/VP,VQ,VP1(80),VQ1(80),VP2(80),VQ2(80)
      COMMON/D/NDZ,G,B,NDZT,Y
      IX=0
      IB=8
      IF(LEVEL-2)31,32,32
   32 CONTINUE
      WRITE(3,2500)R,RN,RZ,X,XN,XZ
   31 CONTINUE
      IF(FAULT-2)21,24,21
   21 CONTINUE
      READ(23'3)NDZ
      CALL NAM(30,31)
      II=50
      IMAX=600
      I=0
      DO 70 K=1,N
      IA2(K)=0
      IF(NDZ(K))9,28,28
    9 VP(K)=0.0
      GO TO 70
   28 VP(K)=0.6
   70 VQ(K)=0.0
      DO 4 K=1,N
      READ(30'K)G
      DO 4 J=K,N
    4 Y(K,J)=G(J)
       DO 14 K=2,N
      READ(31'K)B
      Y(N+1,K)=B(K)
      KT=K-1
      DO 14 J=1,KT
   14 Y(K,J)=B(J)
      READ(31'1)B
      Y(N+1,1)=B(1)
      VP(FBUS)=1.0
      NDZ(FBUS)=0
   11 Z=1
      READ(23'3)NDZT
      NDZT(FBUS)=0
      K=0
  122 K=K+1
      IF(NDZT(K))5,7,6
    5 Z=Z+1
      GO TO 7
    6 CONTINUE
       IP=0.
      IQ=0.
      IF(K-1)201,202,201
  201 KT=K-1
      DO 204 M=1,KT
```

```
      IP=(IP+Y(M,K)*VP(M)-Y(K,M)*VQ(M))
      IQ=(IQ+Y(K,M)*VP(M)+Y(M,K)*VQ(M))
  204 CONTINUE
       IF(K-N)202,205,205
  202 KT=K+1
      DO 205 M=KT,N
      IP=(IP+Y(K,M)*VP(M)-Y(M,K)*VQ(M))
      IQ=(IQ+Y(M,K)*VP(M)+Y(K,M)*VQ(M))
  205 CONTINUE
      IP=IP+Y(K,K)*VP(K)-Y(N+1,K)*VQ(K)
      IQ=IQ+Y(N+1,K)*VP(K)+Y(K,K)*VQ(K)
      P=IP*VP(K)+IQ*VQ(K)
      Q=IP*VQ(K)-IQ*VP(K)
      DP=-P
      DQ=-Q
      CALL ACCL1(IB,IX,I)
      IF(I-IMAX)16,17,18
   18 CALL EXIT
   17 DO 19 M=1,N
      WRITE(3,20)VP(M),VQ(M),M
   20 FORMAT(2F12.5,I10)
   19 CONTINUE
   16 CONTINUE
      Y1=Y(K,K)*VP(K)
      Y2=Y(N+1,K)*VP(K)
       Y3=Y(K,K)*VQ(K)
       Y4=Y(N+1,K)*VQ(K)
      A=IP+Y1+Y4
       AB=IQ-Y2+Y3
      C=Y3-Y2-IQ
       D=IP-Y1-Y4
      IF((ABS(A*D)+ABS(AB*C))-1.0E-6)367,369,369
  367 VP(K)=0.0
      VQ(K)    =0.0
      DVP=0.0
      DVQ=0.0
      IA2(K)=K
       GO TO 8
  369 CONTINUE
       DVP=(D*DP-AB*DQ)/(A*D -AB*C)
       DVQ=(-C*DP+A*DQ)/(A*D-AB*C)
      IF(ABS(DVP)-TOL)1,1,3
    1 IF(ABS(DVQ)-TOL)8,8,3
    8 Z=Z+1
    3 CONTINUE
      VP(K)=VP(K)+DVP
      VQ(K)=VQ(K)+DVQ
    7 CONTINUE
      IF(K-N)122,10,10
   10 I=I+1
      IF(I.LT.II)GO TO 36
      II=II+50
      DO 37 K=1,N
   37 WRITE(3,38)K,VP(K),VQ(K),I
   38 FORMAT(I5,2E20.6,I15,5X, 11HK   VPZ   VQ7 )
   36 CONTINUE
      IF(Z-N)11,12,12
   12 CONTINUE
```

```
      WRITE(3,13)I
   13 FORMAT(1H1,///,I4,6X,'ITERATIONS FOR THE',
     2 ' ZERO SEQUENCE NETWORK',
     1 ///,10X, 'THE ZERO SEQUENCE IMPEDANCE IS' )
      K=FBUS
      IPZ=0.
      IQZ=0.
      IF(K-1)301,302,301
  301 KT=K-1
      DO 304 M=1,KT
      IPZ=(IPZ+Y(M,K)*VP(K)-Y(K,M)*VQ(M))
      IQZ=(IQZ+Y(K,M)*VP(M)+Y(M,K)*VQ(K))
  304 CONTINUE
      IF(K-N)302,305,305
  302 KT=K+1
      DO 305 M=KT,N
      IPZ=(IPZ+Y(K,M)*VP(M)-Y(M,K)*VQ(M))
      IQZ=(IQZ+Y(M,K)*VP(M)+Y(K,M)*VQ(M))
  305 CONTINUE
      IPZ=IPZ+Y(K,K)*VP(K)-Y(N+1,K)*VQ(K)
      IQZ=IQZ+Y(N+1,K)*VP(K)+Y(K,K)*VQ(K)
      RZ=XMOD(IPZ,IQZ)
      XZ=-XMOD(IQZ,IPZ)
      WRITE(3,15)RZ,XZ
   15 FORMAT(1H+,50X,F8.4,2X,'+',2X,'J(',
     1 2X,F8.4,' )')
      IF(LEVEL-2)33,34,34
   34 CONTINUE
      DO 110 M=1,N
      WRITE(3,111)VP(M),VQ(M) ,M
  111 FORMAT(2E30.6,I5)
  110 CONTINUE
      WRITE(3,383)
  383 FORMAT(1H1)
   33 CONTINUE
      WRITE(15'11)VP
      WRITE(15'12)VQ
      GO TO(23,24,25),FAULT
   23 RT=3.0*R+RN+RZ
      XT=3.0*X+XN+XZ
   27 GF=XMOD(RT,XT)
      BF=-XMOD(XT,RT)
      IF(LEVEL-4)40,41,41
   41 CONTINUE
 2500 FORMAT(8E14.6)
      WRITE(3,2500)R,RN,RZ,X,XN,XZ,RT,XT
      WRITE(3,111)R,X,FBUS
   40 CONTINUE
      WRITE(3,111)GF,BF,FBUS
      GO TO 26
   24 RT=R+RN
      XT=X+XN
      GO TO 27
   25 RT=3.0*R+RZ
      XT=3.0*X+XZ
      GF=XMOD(RT,XT)+XMOD(RN,XN)
      BF=-XMOD(XT,RT)-XMOD(XN,RN)
   26 CONTINUE
```

```
      READ(6'FBUS)G
      READ(7'FBUS)B
      G(FBUS)=G(FBUS)+GF
      B(FBUS)=B(FBUS)+BF
      WRITE(6'FBUS)G
      WRITE(7'FBUS)B
      WRITE(23'3)NDZ
      ICT=1
      IF(LEVEL-4)30,43,43
   43 CONTINUE
      WRITE(3,112)GF,BF
  112 FORMAT(2F20.6)
      WRITE(3,368)
  368 FORMAT('THESE VOLTAGES SET TO ZERO',
     1 ' BECAUSE OF THE 0/0 CONDITION')
       DO 380 K=1,N
      IF(IA2(K))380,30,380
  380 WRITE(3,381)IA2(K) ,NDZ(K),NDZT(K),K
  381 FORMAT(I60,2I10,I5,'  IA2  NDZ  NDZT  K')
   30 CONTINUE
      RETURN
      END
```

END OF SEGMENT, LENGTH 1732, NAME ZSSOL

```
      SUBROUTINE PSNS1
      INTEGER BR,FBUS,FAULT,ND(80)   ,TF
       REAL MODV(80)
       DIMENSION PS(80),QS(80),VP(80),VQ(80)
      DIMENSION G(80),B(80)
       COMMON N,BR,L,TF,FBUS,FAULT,RN,XN,RZ-XZ,R,X,BF,GF
       COMMON ICT,IEMF,TOL,LEVEL
      COMMON/D/MODV,PS,QS,VP,VQ,G,B
      COMMON/C10/ISCIND,NCOUNT
      COMMON/C11/NODESC(20)
      COMMON/13/K3
      COMMON/T2/CARD(10)
      CALL DEFBUF(29,80,CARD)
      IF(ISCIND)10,10,3
   10 CONTINUE
      CALL FREAD
      READ(29,93)R,X
   93 FORMAT(2F10.4)
      WRITE(3,7)FAULT,FBUS,R,X
    7 FORMAT(1H1,2I3,2F8.4,'FAULT TYPE  FAULT BUS ',
     1 ' FAULT IMPEDANCE')
      IF(FAULT)3,4,3
    4 STOP
    3 CONTINUE
      DO 5 K=1,N
        READ(6'K)G
      READ(7'K)B
      WRITE(30'K)G
       WRITE(31'K)B
      WRITE(19'K)G
      WRITE(20'K)B
    5 CONTINUE
      READ(10'1)PS
      READ(10'2)QS
      READ(15'1)VP
      READ(15'2)VQ
      READ(23'1)ND
      DO 1M=2,N
      READ(6'M)G
      READ(7'M)B
      VMOD=VP(M)**2+VQ(M)**2
      B(M)=B(M)+QS(M)/VMOD
      G(M)=G(M)-PS(M)/VMOD
       PS(M)=0.0
      QS(M)=0.0
      MODV(M)=0.
      WRITE(6'M)G
      WRITE(7'M)B
      IF(ND(M))1,1,2
    2 CONTINUE
      WRITE(19'M)G
      WRITE(20'M)B
    1 CONTINUE
      WRITE(23'1)ND
      WRITE(10'3)MODV
      WRITE(10'1)PS
      WRITE(10'2)QS
      IF(ISCIND)13,13,15
```

```
   13 CONTINUE
      CALL FREAD
      READ(29,8)ICCNS
    8 FORMAT(I3)
      WRITE(3,18)ICCNS
   18 FORMAT( I6,  ' THE NUMBER OF NEGATIVE SEQUENCE MODS')
      IF(ICCNS)9,11,9
    9 CONTINUE
      DO 11 J=1,ICCNS
      CALL FREAD
      READ(29,70)K,M,R1,X1
   70 FORMAT(2I3,2F10.4)
      JJ=0
      GN = XMOD(R1,X1)
       BN=-XMOD(X1,R1)
      WRITE(3,6)K,M,GN,BN
    6 FORMAT(//,2I6,2E20.6,5X,'NEGATIVE SEQUENCE MODS')
   14 READ(19'K)G
      READ(20'K)B
      G(K)=G(K)+G(M)+GN
      B(K)=B(K)+B(M)+BN
      G(M)=-GN
      B(M)=-BN
      WRITE(19'K)G
      WRITE(20'K)B
      IF(JJ)12,12,11
   12 KT=K
      K=M
      M=KT
      JJ=1
      GO TO 14
   11 CONTINUE
   15 CONTINUE
      RETURN
      END

ND OF SEGMENT, LENGTH   626, NAME  PSNS1
```

```
      SUBROUTINE VFCTR
      INTEGER FBUS,FAULT ,BB,TF
      REAL IPF,IQF
       DIMENSION VP(80),VQ(80),VPN(80),VQN(80) .
      DIMENSION VPZ(80),VQZ(80)
      DIMENSION MPS(3,80)
      DIMENSION A2(10)
       COMMON N,BB,L,TF,FBUS,FAULT,RN,XN,RZ.XZ,R,X,BF,GF
      COMMON ICI,IEMF,TOL,LEVEL
      COMMON/B/A2
      COMMON/T4/IPRINT
      COMMON/D/VP,VQ,VPN,VQN,VPZ,VQZ
      READ(23'4)MPS
      RRN=RN
      XXN=XN
      J=5
      J1=6
      J2=9
      J3=10
      L=0
      READ(15'1)VP
      READ(15'2)VQ
    2 CONTINUE
      READ(15'J2)VPN
      READ(15'J3)VQN
      IF(FAULT-2)4,4,5
    4 IPF= (VP(FBUS)*GF-VQ(FBUS)*BF   )
      IQF= (VP(FBUS)*BF+VQ(FBUS)*GF       )
      IF(LEVEL-4)26,27,27
   27 CONTINUE
      WRITE(3,25)IPF,IQF
   25 FORMAT(///,2E20.5,5X,'IPF AND IQF'///)
   26 CONTINUE
      VNP=-(IPF*RN-IQF*XN)
      VNQ=-(IPF*XN+IQF*RN)
      IF(FAULT-2)6,57,57
   57 VNP=-VNP
      VNQ=-VNQ
      GO TO 6
    5 CONTINUE
       VNP=VP(FBUS)
       VNQ=VQ(FBUS)
    6 CALL LMT(VNP,VNQ,VMOD,THETA)
      DO 1 K=1,N
      S=VPN(K)
      T=VQN(K)
      CALL LMT(S,T,U,W)
      DELTA=THETA+W
      VPN(K)=VMOD*U*COS(DELTA)
       VQN(K)=VMOD*U*SIN(DELTA)
    1 CONTINUE
      WRITE(15'J)VPN
      WRITE(15'J1)VQN
      RN=RZ
      XN=XZ
      J=7
      J1=8
      J2=11
```

```
      J3=12
      L=L+1
      IF(L-2)3,8,3
    3   IF(FAULT-2)2,1000,7
 1000 CONTINUE
       DO 101 M=1,N
      VPZ(M)=0.0
      VOZ(M)=0.0
  101 CONTINUE
      WRITE(15'7)VPZ
       WRITE(15'8)VQZ
      WRITE(15'11)VPZ
      WRITE(15'12)VQZ
      GO TO 8
    7 CONTINUE
      READ(15'J2)VPN
      READ(15'J3)VQN
      RT=3.0*R+RZ
      XT=3.0*X+XZ
      GF=XMOD(RT,XT)
      BF=-XMOD(XT,RT)
      GO TO 4
    8 CONTINUE
      RN=RRN
      XN=XXN
      READ(15'5)VPN
      READ(15'6)VQN
      CALL ROTAT(VPN,VQN,MPS,-1)
      WRITE(15'5)VPN
      WRITE(15'6)VQN
      READ(15'7)VPZ
      READ(15'8)VOZ
      IF(LEVEL-2)30,31,31
   31 WRITE(3,33)
   33 FORMAT(5X,'SEQUENCE VOLTAGES',//,13X,
     2 ' POSITIVE',22x,'NEGATIVE',22X
     1,'ZERO',14X,'NODE')
      DO 32 M=1,N
      WRITE(3,21)VP(M),VQ(M),VPN(M),VQN(M),VPZ(M),VOZ(M),M
   32 CONTINUE
   21 FORMAT(6E15.4,I6)
   30 CONTINUE
      IF(IPRINT.LT.2)RETURN
      WRITE(3,9)
    9 FORMAT(1H1)
      WRITE(3,87)A2
   87 FORMAT(///,5X,10A8,///)
      GO TO(14,15,16),FAULT
   14 WRITE(3,11)FBUS
      GO TO 17
   15 WRITE(3,12)FBUS
      GO TO 17
   16 WRITE(3,13)FBUS
   11 FORMAT(5X,'LINE TO EARTH FAULT ON PHASE A AT BUS',I4)
   12 FORMAT(5X,'LINE TO LINE FAULT ON',
     1' PHASES B AND C AT BUS',I4)
   13 FORMAT(5X,'LINE TO LINE TO EARTH FAULT ON',
     1 ' PHASES B AND C AT BUS',I4)
```

```
   17 CONTINUE
      WRITE(3,10)
   10 FORMAT(////,5X,'NODAL PHASE VOLTAGES'.////,16X,
     2 ' NODE',16X,'PHASE A'
     3,22X,'PHASE B',23X,'PHASE C',///,
     4 16X,3(16X,'MOD       ANGLE'))
      RETURN
      END
```

END OF SEGMENT, LENGTH   733, NAME VFCTR

```
      SUBROUTINE VCOMP(IC)
      INTEGER BB,TF,FBUS,FAULT
      DIMENSION VP(80),VQ(80),VPN(80),VQN(80),VPZ(80),VQZ(
       COMMON N,BB,L,TF,FBUS,FAULT,RN,XN,RZ-XZ,R,X,BF,GF ,I
      COMMON/D/VP,VQ,VPN,VQN,VPZ,VQZ
      COMMON/T4/IPRINT
      READ(15'1)VP
      READ(15'2)VQ
      READ(15'5)VPN
      READ(15'6)VQN
      READ(15'7)VPZ
      READ(15'8)VQZ
      DO1K=1,N
      VAP=VP(K)+VPN(K)+VPZ(K)
      VAQ= VQ(K)+VQN(K)+VQZ(K)
      CALL LMT(VAP,VAQ,VAMOD,R1)
      VBP=VPZ(K)-0.5*(VP(K)+VPN(K))+(SQRT(3.0)/2.0)
     1 *(VQ(K)-VQN(K))
       VBQ=VQZ(K)-0.5*(VQ(K)+VQN(K))-(SQRT(3.0)/2.0)
     1 *(VP(K)-VPN(K))
      CALL LMT(VBP,VBQ,VBMOD,R2)
      VCP=VPZ(K)-0.5*(VP(K)+VPN(K))+(SQRT(3.0)/2.0)
     1 *(VQN(K)-VQ(K))
      VCQ=VQZ(K)-0.5*(VQ(K)+VQN(K))+(SQRT(3.0)/2.0)
     1 *(VP(K)-VPN(K))
      CALL LMT(VCP,VCQ,VCMOD,R3)
      A1=VAMOD*COS(R1)
      A2=VAMOD*SIN(R1)
      A3=VBMOD*COS(R2)
      A4=VBMOD*SIN(R2)
      A5=VCMOD*COS(R3)
      A6=VCMOD*SIN(R3)
      VQ(K)=R1
      IF(R1.LT.0.0001)VQ(K)=0.0
      VQN(K)=R2
      IF(R2.LT.0.0001)VQN(K)=0.0
      VQZ(K)=R3
      IF(R3.LT.0.0001)VQZ(K)=0.0
      VP(K)=VAMOD
      VPN(K)=VBMOD
      VPZ(K)=VCMOD
      R1=R1*57.296
       R2=R2*57.296
      R3=R3*57.296
      A=VAMOD+VBMOD+VCMOD
      IF(A.GT.0.0001)GO TO 8
      VP(K)=0.0
      VPN(K)=0.
      VPZ(K)=0.
      GO TO 1
    8 IF(IPRINT.LT.2)GO TO 1
      WRITE(3,2)K,VAMOD,R1,VBMOD,R2,VCMOD,R3
    2 FORMAT( 11X,I6,1X,3(10X,F10.5,F10.2))
      WRITE(3,6)K,A1,A2,A3,A4,A5,A6
    6 FORMAT(I4,24X,2F10.5,10X,2F10.5,10X,2F10.5,//)
    1 CONTINUE
      IF(IC-2)3,4,5
    3 K1=16
```

```
      K2=17
      K3=18
      K4=25
      K5=26
      K6=27
      GO TO 7
    4 K1=19
      K2=20
      K3=21
      K4=28
      K5=29
      K6=30
      GO TO 7
    5 K1=22
      K2=23
      K3=24
      K4=31
      K5=32
      K6=33
    7 WRITE(15'K1)VP
      WRITE(15'K2)VPN
      WRITE(15'K3)VPZ
      WRITE(15'K4)VQ
      WRITE(15'K5)VQN
      WRITE(15'K6)VQZ
      RETURN
      END

END OF SEGMENT, LENGTH   722, NAME  VCOMP
```

```
      SUBROUTINE EMF(IEMF)
      REAL IP,IQ
      DIMENSION VP(80),VQ(80),PS(80),QS(80)
      DIMENSION A2(2,80),IK(80)
      COMMON/D/A2,IK,VP,VQ,PS,QS
      COMMON/T3/K3
      COMMON/T2/CARD(10)
      CALL DEFBUF(29,80,CARD)
      READ(15'1)VP
      READ(15'2)VQ
      READ(10'1)PS
      READ(10'2)QS
      WRITE(3,1)
    1 FORMAT(/////)
      WRITE(3,7)
    7 FORMAT(19X,'A   + JB',18X,'MOD',8X,'ANGLE',7X,'NODE',
     1 5X,'R   AND   X')
      DO 4 I=1,IEMF
      CALL FREAD
      READ(29,5)K,IC,R,X,P,Q
    5 FORMAT(2I3,4F10.4)
      IF(P)11,12,11
   12 IF(Q)11,13,11
   13 P=PS(K)
      Q=QS(K)
   11 IP=P*XMOD(VP(K),VQ(K))+Q*XMOD(VQ(K),VP(K))
      IQ=P*XMOD(VQ(K),VP(K))-Q*XMOD(VP(K),VQ(K))
      VA=IP*R-IQ*X
      VB=IP*X+IQ*R
      V1=VP(K)+VA
      V2=VQ(K)+VB
   20 FORMAT( ///,  ' VP(K),VQ(K),IP,IQ,VA,VB  ',// 6E17.6,I5)
      WRITE(3,20)VP(K),VQ(K),IP,IQ,VA,VB,K
      CALL LMT(V1,V2,A1,B1)
      B1=57.2958*B1
      WRITE(3,6)V1,V2,A1,B1,K,R,X
    6 FORMAT(7X,2F13.7,F17.6,F10.3,I9,2F10.4)
      IK(I)=IC
      A2(1,I)=V1
      A2(2,I)=V2
    4 CONTINUE
      WRITE(3,10)
   10 FORMAT(////,5X, 'NEW BUS NUMBER AND M/C EMFS VP   VQ ' )
      DO 8 I=1,IEMF
    9 FORMAT(I3,2F10.6)
    8 WRITE(3,9)IK(I),A2(1,I),A2(2,I)
      RETURN
      END
```

END OF SEGMENT, LENGTH   372, NAME EMF

```
      SUBROUTINE ICOMP(IC)
      INTEGER BB,TF,FBUS,FAULT
      DIMENSION VP(240),VQ(240),VPN(240),VQN(240)
      DIMENSION VPZ(240),VQZ(240)
      DIMENSION IBUS(2,240)
       COMMON N,BB,L,TF,FBUS,FAULT,RN,XN,RZ.XZ,R,X,BF,GF ,ICT
      COMMON/D/VP,VQ,VPN,VQN,VPZ,VQZ,IBUS
      COMMON/T4/IPRINT
   70 FORMAT(1H1,///,5X,'LINE CURRENTS',////,
     1 11X,'LINE',21X,'PHASE A',23X,'PHASE B',23X,
     3'PHASE C',////,
     2 17X,3(14X,'MODULUS     ANGLE'))
      IF(IPRINT.LT.2)GO TO 9
      WRITE(3,70)
    9 READ(37'1)IBUS
      READ(24'1)VP
      READ(24'2)VQ
      READ(24'3)VPN
      READ(24'4)VQN
      READ(24'5)VPZ
      READ(24'6)VQZ
      DO 1 K=1,BB
      VAP=VP(K)+VPN(K)+VPZ(K)
      VAQ= VQ(K)+VQN(K)+VQZ(K)
      CALL LMT(VAP,VAQ,VAMOD,R1)
      VBP=VPZ(K)-0.5*(VP(K)+VPN(K))+(SQRT(3.0)/2.0)
     1 *(VQ(K)-VQN(K))
       VBQ=VQZ(K)-0.5*(VQ(K)+VQN(K))-(SQRT(3.0)/2.0)
     1 *(VP(K)-VPN(K))
      CALL LMT(VBP,VBQ,VBMOD,R2)
      VCP=VPZ(K)-0.5*(VP(K)+VPN(K))+(SQRT(3.0)/2.0)
     1 *(VQN(K)-VQ(K))
      VCQ=VQZ(K)-0.5*(VQ(K)+VQN(K))+(SQRT(3.0)/2.0)
     1 *(VP(K)-VPN(K))
      CALL LMT(VCP,VCQ,VCMOD,R3)
      A1=VAMOD*COS(R1)
      A2=VAMOD*SIN(R1)
      A3=VBMOD*COS(R2)
      A4=VBMOD*SIN(R2)
      A5=VCMOD*COS(R3)
      A6=VCMOD*SIN(R3)
      VQ(K)=R1
      VQN(K)=R2
      VQZ(K)=R3
      R1=R1*57.296
       R2=R2*57.296
      R3=R3*57.296
      VP(K)=VAMOD
      VPN(K)=VBMOD
      VPZ(K)=VCMOD
      A= VAMOD+VBMOD+VCMOD
      IF(A-0.0001)1,1,8
    8 IF(IPRINT.LT.2)GO TO 1
      WRITE(3,2)IBUS(1,K),IBUS(2,K),VAMOD,R1,VBMOD,
     1 R2,VCMOD,R3
    2 FORMAT(5X,2I6, 1X,3(10X,F10.5,F10.2))
      WRITE(3,6)A1,A2,A3,A4,A5,A6
    6 FORMAT(28X,3(2F10.5,10X),//)
```

```
      1 CONTINUE
        IF(IC-2)3,4,5
      3 K1=3
        K2=4
        K3=5
        K4=K1+9
        K5=K2+9
        K6=K3+9
        GO TO 7
      4 K1=6
        K2=7
        K3=8
        K4=K1+9
        K5=K2+9
        K6=K3+9
        GO TO 7
      5 K1=9
        K2=10
        K3=11
        K4=K1+9
        K5=K2+9
        K6=K3+9
      7 WRITE(37'K1)VP
        WRITE(37'K2)VPN
        WRITE(37'K3)VPZ
        WRITE(37'K4)VQ
        WRITE(37'K5)VQN
        WRITE(37'K6)VQZ
        RETURN
        END

ND OF SEGMENT, LENGTH   692, NAME  ICOMP
```



```
        SUBROUTINE CRNT(I5,I1,I2)
```

```
      SUBROUTINE CRNT(I5,I1,I2)
      INTEGER BB
         DIMENSION VP(80),VQ(80)     ,G(80),B(80)
      DIMENSION CIP(240),CIQ(240),IBUS(2,240)
      COMMON N,BB
      COMMON/D/VP,VQ,G,B,CIP,CIQ,IBUS
      COMMON/D1/IC(10)
      READ(37'1)IBUS
      READ(15'I5)VP
      I6=I5+1
      READ(15'I6)VQ
      IX=1
    1 M=IBUS(1,IX)
      READ(I1'M)G
      READ(I2'M)B
    2 I=IBUS(2,IX)
      V1=VP(M)-VP(I)
      V2=VQ(M)-VQ(I)
      CIP(IX) = V2*B(I)-V1*G(I)
      CIQ(IX)=-(V1*B(I)+V2*G(I))
      CIR=SQRT(CIP(IX)**2+CIQ(IX)**2)
      IF(IC(5).EQ.1)GO TO 10
C       IF LOAD FLOW ANALYSIS, PRINT ACTAUL VALUES.
      IF(CIR.GT.0.2)GO TO 10
      CIP(IX)=0.0
      CIQ(IX)=0.0
   10 CONTINUE
      IF(IX-BB)3,4,3
    3 IX=IX+1
      IF(M-IBUS(1,IX))1,2,1
    4 CONTINUE
      IF(I1-19)5,6,7
    5 J = 1
      K=2
      GO TO 8
    6 J=3
      K=4
      GO TO 8
    7 CONTINUE
      J=5
      K=6
    8 CONTINUE
      WRITE(24'J)CIP
      WRITE(24'K)CIQ
       RETURN
      END

.ND OF SEGMENT, LENGTH   320, NAME  CRNT
```

```
      SUBROUTINE NAM(K,M)
      INTEGER BB,TF,FBUS,FAULT
      DIMENSION G(80),B(80)
      DIMENSION NDZ(80)
      COMMON N,BE,L,TF,FBUS,FAULT,RN,XN,RZ,XZ,R,X,BF,GF
      COMMON ICT,IEMF,TOL,LEVEL
      COMMON/D/NDZ,G,B
      IF(LEVEL)15,15,14
   14 WRITE(3,1)
    1 FORMAT(1H1)
      IF(K-19)2,3,4
    2 WRITE(3,5)
    5 FORMAT(2X,'  POSITIVE')
      GO TO 6
    3 WRITE(3,7)
    7 FORMAT(2X,'  NEGATIVE')
      GO TO 6
    4 WRITE(3,8)
    8 FORMAT(2X,'      ZERO')
    6 WRITE(3,13)
   13 FORMAT(1H+,13X,'SEQUENCE NODAL ADMITIANCE MATRIX',///)
      DO 9 J=1,N
      READ(K'J)G
      READ(M'J)B
      DO 10 I=1,N
      IF(G(I))16,17,16
   17 IF(B(I))16,18,16
   16 CONTINUE
      WRITE(3,11) J,I,G(I),B(I)
   11 FORMAT(50X,2I6,2F20.5)
   18 CONTINUE
   10 CONTINUE
      WRITE(3,12)
   12 FORMAT(//)
    9 CONTINUE
   15 CONTINUE
      RETURN
      END
```

ND OF SEGMENT, LENGTH   177, NAME NAM

```
      SUBROUTINE ROT(A,B,M)
      C=(A*SQRT(3.0)/2.0)+(0.5*B*(-1**(M+1)))
      D=(B*SQRT(3.0)/2.0)+(0.5*A*(-1**M))
      A=C
      B=D
      RETURN
       END

END OF SEGMENT, LENGTH    83, NAME  ROT
```

```
      FUNCTION XMOD(A,B)
      IF(A)1,2,1
    2 XMOD=0.0
      RETURN
    1 XMOD=A/(A**2+B**2)
      RETURN
      END

END OF SEGMENT, LENGTH    53, NAME  XMOD
```

```
      SUBROUTINE  LMT(A,B,C,ANGLE)
C     THIS SUBROUTINE FINDS THE MODULUS AND ANGLE FOR VECTORS
C     BETWEEN 0 AND 360. GIVEN THE CARTESIAN CO-ORDS A AND B.
      C=SQRT(A**2+B**2)
      ANGLE=0.0
      IF(ABS(A)-0.001)2,9,9
    9 IF(A)1,3,3
    3 IF(B)4,5,5
    4 ANGLE= 6.2831852
      GO TO 5
    1 ANGLE= 3.1415926
    5 ANGLE= ANGLE+ATAN(B/A)
      RETURN
    2 IF(ABS(B)-0.001)8,10,10
   10 IF(B)6,7,7
    6 ANGLE=4.7123889
      RETURN
    7 ANGLE= 1,5707963
    8 RETURN
      END

END OF SEGMENT, LENGTH    142, NAME  LMT
```

```
      SUBROUTINE ROTAT(VP,VQ,MPS,N2)
      INTEGER BB,TF,FBUS
      DIMENSION VP(80),VQ(80),MPS(3,80)
      COMMON N,BB,L,TF,FBUS
      IF(FBUS)3,3,4
    3 IT=0
      GO TO 5
    4 IT=MPS(2,FBUS)
    5 CONTINUE
      DO 1 M=1,N
      MPS(3,M)=MPS(2,M)-IT
      CALL LMT(VP(M),VQ(M),A,B)
      C=0.523599*MPS(3,M)*N2
      B=B+C
      VP(M)=A*COS(B)
    1 VQ(M)=A*SIN(B)
      RETURN
      END

END OF SEGMENT, LENGTH   155, NAME  ROTAT
```