CHAPTER IV

DESCRIPTION OF THE DEVELOPED PROGRAMS

4.1. Introduction

The programs developed are written in a codal language called Fortran I. The programs are punched in decks of cards for loading an I.B.M. 1620 electronic digital computer. The Fortran programs are called source programs. They are not recognized by the computer. A card deck of processor is used to translate Fortran language into Machine Language which is recognized by the computer. A translated card decks is called an object program. The object deck, being loaded into the computer with a subroutine, can instruct the computer to operate according to the program. Data are to be punched on cards. The solutions may be punched on cards and /or typed on paper.

Due to the limited number of core storage of the available I.B.M. 1620 computer, all steps of computation for either a three-phase short circuit analysis or an one-phase short circuit analysis cannot be developed integrally in one big program. In this investigation, six small programs are developed. They are

(a) Program I for forming a nodal admittance matrix of a system.

(b) Program II for inverting a nodal admittance matrix to a nodal impedance matrix.

(c) Program III for modifying a nodal impedance matrix.

(d) Program IV for calculating a prefault voltage distribution.

(e) Program V for calculating fault current and voltage distributions due to three-phase short circuit at various busbars.

(f) Program VI for calculating fault current distributions due to one-phase short circuits at various busbars.

The sequence of the programs for a three-phase short circuit analysis is

(a) Program I

(b) Program II

(c) Program III (if required)

(d) Program IV (if required)

(e) Program V

In the analysis of one-phase short circuit, Program VI replaces Program V in the above sequence.

In either analysis, the biggest system that can be handled on an I.B.M. 1620 computer with 20K core storage must not have more than fifteen busbars. This limitation is due to the great consumption in core storage of the matrix inversion program.

4.2. Program for Nodal Admittance Matrix Formation.

The program forms a nodal admittance matrix of a system from the generator and load data, and from the branch data. Impedance values are converted to admittance values; loads, to equivalent admittances. All elements of the matrix are first stored as zero. Shunt admittances, load equivalent admittances and generator internal admittances are directly added to the appropriated diagonal elements. In case of transformers and trnasmission lines connecting between busbars i and j, using a generalized π-network representation as described in (2.4), the admittance added to the diagonal element Y_{ii} is

$$nY + n(n-1)Y + b \quad = \quad n^2Y + b$$

the admittance added to the diagonal element Y_{jj} is

$$nY + (1-n)Y + b \quad = \quad Y + b$$

the admittance added to the off-diagonal Y_{ij} is

$$-nY$$

Being a symmetrical square matrix, only the upper triangular half of the matrix including all diagonal elements is stored in the computer. Each element of the matrix is referred to by a subscript K. K is relzted to the row number i and the column number j by

$$K = i(i - 1)/2 + j \quad \text{for i is greater than or equal to j} \qquad (4.1)$$

and $$K = j(j - 1)/2 + i \quad \text{for i is smaller than j} \qquad (4.2)$$

The outputs produced sequentially from the program are

(a) equivalent current sources and shunt admittances of generators.and loads, for loads current sources are zero.

(b) the upper triangular half of the matrix formed.

(c) the prefault voltages expressed in rectangular form. They are punched only when program switch number 3 on the console is ON.

The details of the program may be studied from the flow chart and program body in Appendix A.

4.3. Program for Matrix Inversion.

The program is used to invert a modal admittance matrix to a nodal impedance matrix. The method used is the elimination method which is performed in the following manner.

Let the elements of the nodal impedance matrix be:

$$\begin{array}{ccccc}
Y_{11} & Y_{12} & Y_{13} & \ldots\ldots\ldots\ldots\ldots & Y_{1n} \\
Y_{21} & Y_{22} & Y_{23} & \ldots\ldots\ldots\ldots\ldots & Y_{2n} \\
Y_{31} & Y_{32} & Y_{33} & \ldots\ldots\ldots\ldots\ldots & Y_{3n} \\
. & . & . & & . \\
. & . & . & & . \\
. & . & . & & . \\
. & . & . & & . \\
Y_{n1} & Y_{n2} & Y_{n3} & \ldots\ldots\ldots\ldots\ldots & Y_{nn}
\end{array}$$

The method begins by a selecting of the biggest pivot element in the first column. Then a unit vector, which contains a unity in the pivot row and zeros elsewhere, is stored adjacent to the last column. Then the following steps are performed.

(a) For the pivot row elements, a new value is computed from

$$Y'_{pj} = \frac{Y_{pj}}{Y_{p,1}} \qquad j = 1,\ 2,\ \ldots\ldots\ n+1 \qquad (4.3)$$

(b) For, all other elements, a new value is

$$Y'_{ij} = Y_{ij} - (Y_{ip})(Y'_{pj}) \qquad i = 1,2\ldots.\ n \neq p \qquad (4.4)$$

$$j = 1,2\ldots\ldots n+1$$

(c) The added unit vector in the n+1 column now becomes a new vector. It is then transferred to the column which contains the pivot element.

(d) The next column is searched for a biggest pivot element which is not in any previous pivot rows. A unit vector is again added to the n+1 column, and steps (a), (b) and (c) are repeated again.

The above four steps are repeated for n times to inverse the original nodal admittance matrix into the nodal impedance matrix.

The idea of selecting the biggest pivot element is to eliminate using a zero as a pivot element. Nevertheless, a part of core storage must be sacrified in registering the pivot row number previously used and in registering the positions of the rows and columns of the nodal impedance matrix which are differed from the original matrix because the selection of a pivot row is not in a sequential order.

The matrix stored in the core storage is a full square matrix. Each element is referred directly by its row and column numbers. The full square matrix in complex number consumes a great deal of core storage. The maximum size of a matrix that can be inverted is 15 rows by 15 columns.

If the program switch No. 2 only is on, the output is the upper triangular half of the matrix; if the program switch No. 3 only is on, only the diagonal elements are punched; and if all program switches are off, the full square matrix is the output.

Details of the program may be studied from the flow chart and the program body in Appendix B.

4.4 Program for Modifying a Nodal Impedance Matrix

The program modifies the nodal impedance matrix when there are changes in shunt impedances, loads, generator internal impedances, line impedances and their shunt susceptances, and tap settings of transformers. It enables the system with new condition created by a line outrage or adding of a line to be investigated from the original nodal impedance matrix.

The method of the modification is based on Kron's formular which is explained briefly in the paper by Day and Parton.(2) The formulae derived may be quoted here

(a) In case of a change in a shunt impedance, a load or a generator at busbar m

Let Y be the admittance being added, then the new nodal impedance matrix is

$$[Z'] = [Z] - \frac{(\text{column m of } [Z])\ (\text{row m of } [Z])}{1/Y + Z_{mm}} \qquad (4.5)$$

(b) In case of a change of any impedance value connecting between busbars m and k

Let Y be the admittance being added, then

$$[Z'] = [Z] - \frac{(\text{col. m-col. k})\ (\text{row m-row k})}{1/Y + Z_{mm} + Z_{kk} - 2Z_{mk}} \qquad (4.6)$$

Since transmission lines and tapped transformers can be represented by π - network, modification due to the changes in line characteristics and in tap settings of transformers can be handled by applying formula (4.5) to the two legs of π- network, and formula (4.6) to its arm.

In this program only the upper triangular half of the matrix is stored. Each element can be referred to by a subcript K as described in 4.2.

In order to make a change in the characteristics of a system element, two data cards are needed.

For load data, one card contains the negative power of the original one; the other, a new required power.

For branch data, the negative of the original impedance and shunt susceptance; the new impedance and shunt susceptance, or a new tap setting.

Console switch No. 2 should be on when new data of generators and loads are loaded, otherwise it is OFF.

The outputs of the program are

(a) Equivalent current sources and internal admittances. They are punched only when console switch No. 2 is on.

(b) The modified nodal impedance matrix in a full square form.

The details of the program can be studied from the flow chart and program body in Appendix C.

4.5. Program for Prefault Voltage Calculation.

The prefault voltage distribution in a system for the short circuit studies may be obtained either from the solution of a load flow study or from the estimated outputs and terminal voltages of generators. In the latter case, this program has to be used to determine the approximate prefault voltage distribution.

The input data of the program are the nodal impedance matrix, the approximate terminal voltages of the generator busbars, and the generator data. The prefault voltage distribution is determined from

$$[V] = [Z] \left[Y_g E_g\right]$$

where $\left[Y_g E_g\right]$ is a vector of equivalent current sources.

A equivalent current source at busbar m is determined from the terminal voltage and generation of a generator connected to the busbar, and is derived in the following manner:

$$E_g = V_m + I_g Z_g \qquad (4.8)$$

$$Y_g E_g = V_m Y_g + I_g$$

$$\text{and } S_g^* = V_m^* I_g \qquad (4.9)$$

$$\text{hence, } Y_g E_g = Y_g V_m + \frac{S_g^*}{V_m^*} \qquad (4.10)$$

The output of this program is only the voltage at each busbar in rectangular form.

Details of the program is shown in the flow chart and program body in Appendix D.

4.6 Program for Fault Cirrent and Voltage Distribution in a System with a Three-Phase Short-Circuit.

This program as the last part of the three-phase short circuit analysis calculates the fault levels at busbars and the distributions of fault currents and fault voltages of the system.

The prefault voltage data is first read into the computer. The fault level at busbar m is determined when a driving point impedance Z_{mm} is read, from equation (3.6)

$$I_f = \frac{V_m}{Z_{mm}}$$

and the fault level is $= V_m I_f^*$ (4.11)

the power for the circuit breaker rating $= CB \times V_m I_f^*$

(4.12)

where CB is a multiplying factor used to take into account the d.c. component of a fault current. It depends on the ratings of a circuit breaker. Values of CB may be looked up in Westinghouse Electric Corporation's 'Electrical Transmission and Distribution Reference Book'$^{(6)}$.

A scalar MVA is obtained by multiplying a per unit power by the base MVA of the system; a scalar current by the base current.

The remaining elements in row m of the nodal impedance matrix are read, and the fault voltages at all busbars are calculated from equation (3.7)

$$V_{K(f)} = V_K - I_f Z_{mK} \quad K = 1, 2, \ldots\ldots\ldots n$$

After having loaded a data of a branch connected between busbars K and l into the computer the fault current flowing from busbar K into the branch is determined from equations.

$$I_{Kl} = (nV_{K(f)} - V_{l(f)})(nY) + bV_{K(f)} \quad (4.13)$$

and
$$I_{lK} = (\frac{1}{n}V_{l(f)} - V_{K(f)})(nY) + bV_{l(f)} \quad (4.14)$$

The current generated by a generator connected to busbar K is determined from the busbar voltage and the generator data

$$I_K = Y_g E_g - Y_g V_{K(f)} \quad (4.15)$$

The output of the program are

(a) Fault level at busbars in ampere currents (arguments in degrees) and in MVA; and MVA for circuit breakers.

(b) Fault currents in any parts of the system; magnitudes in ampere , arguments in degrees.

(c) Fault voltage at busbars; magnitudes in per unit, and argument in degrees.

If the console switch No. 1 is on, only (a) is the output.

The details and more explanations of the program may be looked up in the flow chart and program body as shown in Appendix E.

4.7. Program for Calculation of Fault Current Distribution in a System with a One-Phase Short-Circuit.

This is the last program in an analysis of a system under one-phase short circuit. The prefault voltage distribution in the positive network of the system is first fed in the computer. The fault level at busbar m is determined when the driving point impedance at the busbar of the three sequence networks of phase a are fed in.

Each sequence component of the fault current at the busbar is equal to each other and is determined from equation (3.23)

$$I_{a1} = \frac{V_m}{Z_{mm1} + Z_{mm2} + Z_{mm0}}$$

The phase value of the fault current at the bus is

$$I_f = 3.I_{a1} \qquad (4.16)$$

and the fault level is determined from (4.11) and (4.12).

The remaining elements in row m af each sequence network are fed in the computer. The fault voltage distribution in each sequence network is determined from equations (3.24), (3.25) and (3.26).

Once the fault voltages distribution in each network is calculated, the fault current in any part of the system can be obtained by applying either equation (4.13) and (4.14) or equqtion (4.15) to determine the sequence components. The required current in phase a is the sum of these sequence components.

In this program a qutity of some significance for earth-fault protective relay applications, the equivalent out-of-balance current for the three-phases $3I_o$, is also obtained.

The outputs of the program are

(a) Fault levels in ampere currents (arguments in degrees) and in MVA, and MVA for circuit breaker ratings.

(b) Phase values of fault current in system elements; and three time the phase zero sequence component of the current. Current amplitudes are expressed in ampere and their arguments in degrees.

If the console switch No. 1 is on, only (a) is printed out.

The details of the program may be studied from the flow chart and program body in Appendix E.

4.8 Tests of the Programs

Each program has been tested separately on problems from papers and texbooks to yield accurate results. They are accurated up to the last digit but one of the input data. The set of programs for analysing three phase short circuit has been tested on the example of four busbars in the Departmental Engineering Report No. 602/SD 10,520 of the Associated Electrical Industries Limited to yield results which are accurate up to the last digit but two of the original data. The set of programs for analysing one phase short circuit has also been tested on an example of eleven busbars for the report No. 602/SD 10,524 of the same source to yield results of the same degree of accuracy.