ON SOME ASPECTS OF DIGITAL COMPUTATION IN POWER SYSTEM PROBLEMS

A Dissertation
submitted in partial fulfilment
of the requirements for the Degree
of
MASTER OF ENGINEERING
in
POWER SYSTEM ENGINEERING

By
KRISHNA BEHARI MISRA

DEPARTMENT OF ELECTRICAL ENGINEERING
UNIVERSITY OF ROORKEE
ROORKEE
U.P.
(INDIA)
July, 1966

CERTIFICATE

Certified that the dissertation entitled "On Some Aspects of Digital Computation in Power System Problems " which is being submitted for by Mr. K.B.Misra in partial fulfilment for the award of the degree of Master of Engineering in Power System Engineering of University of Roorkee, is a record of candidates' own work carried out by him under my supervision and guidance. The matter embodied in this dissertation has not been submitted for the award of any other Degree or Diploma.

This is to certify that he has worked for a period of seven months from Dec. '65 to June '66 for preparing dissertation for Master of Engineering Degree at the University.

P Mukhopadhyay
(Dr.P.Mukhopadhyay)
Associate Professor
Deptt. of Electrical Engrg.
University of Roorkee

Roorkee
Dated 30-7-66

ACKNOWLEDGEMENTS

The author is indeed very much grateful to Dr. P. Mukhopadhyay for initiating the topic and for his constant encouragement and guidance.

Thanks are due to Professor C.S. Ghosh , Head of the Electrical Engineering Department, for providing various facilities.

The author is also thankful to Dr. H.C.Misra, Reader in Civil Engineering Department for his help and very much appreciate the cooperation extended by computer Centre staff of S.E.R.C.

ROORKEE
DATED JULY 30, 1966

K.B.MISRA

CONTENTS

SYNOPSIS

During the past decade, Digital Computers have introduced the entirely new approach to the problems, which were otherwise considered almost impossible or had some indirect method of attack.

There has been a great endeavour towards computerising any process that involved considerable labour and time, for the solution of system problems which different method of approach.

The aim of this thesis, is to popularise the use of Digital Computers to the solution of different systems problems encountered; however, this is merely an attempt in this direction.

Several methods exist for the solution of network problems and are suggested almost every year. A comparative study thus is unavoidable and therefore discussed herein.

Recently, Monte Carlo methods have been applied to a variety of problems, hence they have been given a due place in this thesis for the solution of the linear network equations and for the solution of Laplace Equations which in irregular and multidimensional regions, present a difficult situation with the conventional methods.

The Computer programmes are drawn for both the above cases and the result s of the problems chosen have been discussed at par.

Although several problems, such as setting up of self and mutual impedances and Load flow stu ies, usually as a practice by System Engineers at large, are tried on A.C. Network Analyzer, the digital methods for the same are some times superior or to say more convenient as regards time, economy, accuracy and with larger data handling capabilities, and therefore should draw the attention.

The author here discusses different methods of digital approache to the above problems and by taking specific problems examplifies some of the salient features of the same.

A study of optimising acceleration in the iterative process has been illustrated.

The computer programmes for each type of study are chalked out and tested to run successfully.

CHAPTER - 1

1.1 Introduction

It is a well recognised fact that A.C. Network Analyser is an invaluable tool for power system engineers for solving a variety of problems such as Load Flow Studies, short circuit and stability problems. For quite a long time the system engineers were carrying out all above studies on a.c. network analyser only, but with the advent of digital computer, the attention has been more or less diverted towards the use of the computer for solving all those problems which are solved on a.c network analyser.

It has been shown (43) that a digital computer of medium speed and size such as IBM 650, can ~~be~~ compete economically with the network analyser for system of moderate size for Load Flow studies which is more complex problem in nature for its solution. Less cost per compution and increased capacity for handling the system problems seems to be main factors in favour of digital computer. There are two types of problems that have encouraged the use of digital computer.

1. Those problems which were solved by other methods. In this case they only replace the calculation procedure to bring in speed and accuracy.

2. Those problems which were never before attempted because of practical limitations.

The type 1 includes problems such as the calculation of transmission line constants, impedances (2) sag calculations, Load forecasting (6) etc.

The problems, that were solved on analogue computer or a.c network analyzer are now being replaced by digital computer include load flow studies, short circuit calculations (7,8) and transient stability studies (10)

The type (2) problems may be the exact probabilistic determination of generation reserves and the inclusion of transmission losses in the economical dispatch of power. Monte Carlo methods may be put under this catagory. Many areas of application of digital computer to power system problems are not much investigat ed.

The solution of the lightenping problems by exact field theory equations, system design optimisation and other operations research type of studies would yield practical and useful data by digital analysis.

In short, some of the problems for which digital computer may be used to system engineering include:

1. Network impedance calculations
2. Short circuit calculations
3. Load flow studies
4. Stability studies
5. Loss studies
6. Microwave relaying
7. Preparation of impedance data from equipment punched card file.
8. Parametric study in bundle conductor design. In no case the above list may be calld as complete.

The digital computer may find several applicatins with the complete and proper analysis available for the problems.

1.2 Difficulties encountered with Digital Computer Applications

In the begining, the main difficulty in the use of computer was its availability only but this is no problem now when more and more new computers are coming up, differing in size and speed.

As the computer became available the problem rested on the nonavailability of persons computer oriented in thinking.

The problems have to be tried by the persons who know the digital analysis and problem itself. The work envolves, rather a combined effort of a digital analyst, perhaps a mathematician or engineer expert in numerical analysis, computer construction and logic, and the many ramifications and tricks of programming, and that of a power system engineer who knows the problem well.

For the most economical, efficient and successful application of digital computer, the above mentioned abilities must be present in the person who attempts the problems. The difficulty arises when the trained digital analyst is not available and also when the power system engineer is not computer-oriented.

The right appreach would be to ask power system engineer to think in terms of flow charts or logic diagrams. He must be made to learn to organise his problem for computer solution and develop optimum logic for the orderly and economical solution. Finally the system engineer must be well-conversant with the language of the machine also so as to know the complete set up of the whole process of solution.

However, reference (1) suggests the use of computer programme which actually bridges the gap between the Power

System Engineer's language and the computer language. In short, for a satisfactory use of computer the system engineer must first acquire the abilities of a trained digital analyst which is the only difficulty present in the application of digital computer to system problems.

1.3. Comparison of Digital Computer and A.C. Network Analyser

Although the digital computer is finding more and more wide acceptance as regards solution of system problem nevertheless the A.C. Network analyzer still holds the same recognition. This is due to the fact that a.c. network analyser simulates the physical network, the loads and the sources and so aids the system engineer in perceiving the actual situation of the problem. The printed output of digital computer requires interpretation.

There being less chances of human error in recording the data, the digital computer has an advantage in that respect becamse the complete recording of input condition and the information gained in the solution is ensured.

The digital computer is very much economical when few changes from the base case are to be studied on the other hand much time may be wasted on a.c. network analyser. Considerable time may be spent between the runs in interpreting results in order to make decisions as regards what condition must be studied next and so on.

It is easy to handle a system on digital computer with minimum time waste and since the charges are to be made on actual time used on computer, the studies conducted may be quite economical.

In some cases such as loss formula calculations, the accuracy the a.c. network analyser provides is inadequate and hence the use of digital computer is justified, in that it gives desired accuracy.

Another advantage in favour of digital computer is the size of the systems that a large-sized computer can handle. Usually, the a.c. network analyzer do not have enough sources lines and other components to solve an extensive problem. Thus the size of the system is restricted by the use of network analyser where as computer can handle very large systems also (53).

CHAPTER - 2

NETWORK SOLUTION

2.1 Formulation of Network Equations

Apart from the analogue method of solution of network problems the digital methods of solution of network are becoming more predominent with the new techniques of developing the network equations and their solution in general.

Improvements are often suggested in developing and formulating the network equations.

The present chapter is more concerned with the solution of network equations rather than formulation of the same. However in the advent of fast developing techniques a discussion will not be out of place.

Any network can be defined by means of generalised loop current equations or Mesh Equations and Node Equations in the form $\left[E_m\right] = \left[Z^{\circ}\right]\left[I_m\right]$ and $\left[I_n\right] = \left[Y\right]\left[E_n\right]$ respectively Here Z° and Y are the loop or mesh-impedance matrix and the cut-set admittance matrix respectively. The loop-impedance matrix is established by the matrix equation

$$C_t\ Z_b C = Z^{\circ} \qquad \ldots\ldots \qquad (1)$$

where Z_b is the branch impedance matrix and C is one of several possible loop connection matrices. The loop connection matrix and its transpose C_t (or the transpose conjugate if the elements are complex) define the relationships between the primary variables of the networks, branches current and branch voltages and an arbitrary set of secondary variables, loop currents and loop voltages.

In a connected network of n nodes and b branches the following relationships exist

$$n = n_t - 1 \qquad (2)$$

$$m = b - n \qquad (3)$$

where n is the number of independent nodes and m is the number of independent loops.

The relationship between branch currents and an arbitrary set of independent loop current is given by the matrix equation

$$C\ I_m = I_b \qquad (4)$$

I_b is a column vector of b branch current, I_m is a column vector of m independent loop currents and C is a bXm loop connection matrix. The columns of C must be linearly independent and its elements are generally + 1 , 0 and -1 Similarly the node and cut-set admittance matrices could be found by $Y = K\ Y_b K_t$

Where K is node-branch incidence matrix or cutset branch incidence matrix and Y_b is the admittance matrix.

The usual method for obtaining the independent loop equations requires defining the network geometry in terms of a tree and links. Then by the process of closing one ~~xxx~~ link at a time, tracing around the loops thus formed and recording the polarities of branches, encountered, a loop connection matrix is formed.

Similarly by choosing proper cut-sets, the cut-set branch-incidence matrix can also be built up easily.

Ordinarily, it is about equally as difficult to determine a modal connection matrix as it is ~~to~~ a loop connection matrix. There is one exception, the modal connection matrix for the commonly use node-to-datum choice of node-pair voltages is trivially simple to establish. The datum is usually ground and node-pair voltages are the potentials of the separate nodes with respect to ground.

In the formulation of mesh-impedance matrix or cutset admittance matrix the important feature is the setting up of connection matrix and logic steps are required to be performed by the computer.

W.F Tinney (14) in his paper suggested a method of obtaining loop connection matrix using matrix operations instead of geometric logic. It was shown that the concepts of three, links, loops and other topological interpretations are unnecessary for the guidance of computer logic. The main draw back of loop analysis is the difficulty in establising a loop connection matrix.

When the system is large and network is non-planar the usual topological methods are difficult to handle. Nevertheless the trend in digital computer analysis of power system networks has been towards the use of driving point and transfer impedance matrix instead of the inverse loop impedance matrix-this is mainly due to the better methods available for setting them (35, 37).

H.Edelmann(17) suggested a numerical algebraic generation of I pedances and Admittance matrices by Set-Theoritical Intersections. The set-theoretical generation affords two important advantages i.e.,the storage requirements for the intermediate result which compared with those for the end result, are considerable, are omitted. The total computing time for the formation of these matrices is at the same time reduced to approximately one-eighh

2.2 Solution of Network Equations

The formuel tion of Network equation is achieved in the more general form of AX = Y, in which case A is a nonsingular nxn matrix of coefficient either copedances or comittances, x and y are the column vectors of loop currents and loop potential sources in case of Mesh-Impedance equations or of node-potential and Node current sources in case Node Equations. The numbet of independent loop currents in case of mesh-equations and the number of nodes equal to n.
Once the network equations are represented in AX = Y form, the solution can be obtained by either solving this set of linear equation by any conventional method or by a matrix inversion subroutine for inverting the pertinent matrix of A to find the unknown column vector X.

If there are many such sets with the same matrix A and inverse matrix A^{-1} is calculated such that $A^{-1}A = I$ the identity matrix , and for each Y , $X = A^{-1}Y$ is the solution to AX = Y. With the increased reliance on computers to solve large systems of equations, the size of systems considered practical to solve has increased enormously. As the size of these systems has grown to reach the limitations imposed by equipment configurations, more economical methods have been sought. In addition to the economy due to symmetry significant economies in both computer memory and computing time can be achieved by taking advantage of sparsity of A, where it is significant.

In many problems, such as those involving electrical circuits,the matrix A is sparse (more than 90% zero elements) since lines do not connect each bus to every bus (18,20) but A^{-1} is full.

Sato and Tinney (18) describe a method for achieving the same numerical results as multiplication by A^{-1} without actually computing and storing A^{-1}.

Each pivot division, row elimination, and back substitution in a Gaussian elimination corresponds to premultiplying by an elementary matrix. Hence A^{-1} can be represented as a product of elementary matrices A_{ij}, each differring from I only in the ijth entry. The A_{ij} coefficients turn out to be the equation ij coefficients encountered in Gaussian elimination. If the number of the coefficients created in the elimination process is small to n^2, so the array is still sparse, the elementary mattices can be stored as a list including location indicators in much less space than by n by n inverse matrix. In symmetric case this amounts to storing alist of the location and the values of the coefficients in upper triangle. There is a considerable economy in both forming and usig this list as opposed to calculating and using A^{-1}.

N Sato (15) describes the existance of a definite correspondance between the matrices obtained from the mesh and nodal methods, and from that obtains the mesh and nodal inverse matrices by means of elimination scheme the effectiveness of which totally depends upon the network configuration.

The mesh-transformation matrix C of a network can also be calculated by an efficient method.

A row-by-row matrix inversion method employing an iterative process is described in reference (20), and is also given in Appendix G. The advantages of this iterative method of finding inverse are that the labour of finding the whole inverted matrix need not be taken and in some cases a part-solution may be just sufficient. The matrix-inversion process using known methods becomes time consuming because original matrix can not be stored wholly in memory. The row-by-row matrix inversion method allows economic inversion of admittance matrixs of considerable large size than possible with convention methods.

More recently (May 1966) W.F.Tinney (19) has evolved even more simpler method of solving network problems. The method not only applies to the network problems, as he claims but to all systems of linear equations, symmetric or not real or complex, sparse or full.

It has an advantage in that it is much faster and requires very less memory than any other method using impedance matrix or hybrid matrix. It provides direct solution of linear net work problems of order of 1000-2000 in 32 K word memory.

The conventional methods and the Monte Carlo methods of solving a set of linear equations have been discussed as regards their advantages and limitations in reference (33). The theory of matrix inversion by Monte Carlo method has been also dealt in there and for the sake of ready reference a copy of reprint of the same is being attached to the thesis.

2.3 Illustrative Problems

Case of Symmetric matrix

A d.c. network shown in figure 2-1 has been taken as a problem for illustrating the use of different methods for the solution of network equation.

The network, under consideration gives rise to a set of linear equations although with real elements only, nevertheless same approach can also be applied if the elements were complex as is the case with an a.c. network. The iterative method considered here would be modified to allow for complex quantities, or the method of Appendix G can be directly applied to solve.

Formulation of Equations

The method as described in reference (12) is used to form the linear equations.

The node equations would be of the form I = YE or more specifically,

$$I_1 = Y_{11}E_1 + Y_{12}E_2 + Y_{13}E_3 \text{ at node } 1$$

$$I_2 = Y_{21}E_1 + Y_{22}E_2 + Y_{23}E_3 \text{ at node } 2$$

$$I_3 = Y_{31}E_1 + Y_{32}E_2 + Y_{33}E_3 \text{ at node } 3$$

the terminal admittence as determined as

$$\text{Self admittances} \begin{cases} Y_{11} = y_{01} + y_{12} + y_{13} \\ Y_{22} = y_{02} + y_{12} + y_{23} \\ Y_{33} = y_{03} + y_{13} + y_{23} \end{cases}$$

$$\text{mutual admittances} \begin{cases} Y_{12} = Y_{21} = -y_{12} \\ Y_{13} = Y_{31} = -y_{13} \\ Y_{23} = Y_{32} = -y_{23} \end{cases}$$

Applying the above equations to the d.c. network under consideration, of real elements only the equations become

$$\begin{vmatrix} I_1 \\ I_2 \\ I_3 \end{vmatrix} = \begin{vmatrix} 0.4 & -0.2 & -0.1 \\ -0.2 & 0.5 & -0.1 \\ -0.1 & -0.1 & 0.6 \end{vmatrix} \quad \begin{vmatrix} E_1 \\ E_2 \\ E_3 \end{vmatrix}$$

The $[I]$ column vector for which E's are computed has been taken as $\begin{vmatrix} 1 \\ 0 \\ 1 \end{vmatrix}$

The voltages E_1, E_2 and E_3 are found by Itrative and Monte Carlo Method.

A computer programme for the Iterative method of solving linear equations has been written and included in Appendix B . The equation of the network of fig. 2-1 were solved and the results have be listed in Table 2-3-1. The error after each iteration is also given so as to know the accuracy of the result at each iteration.

The values of the unknowns i.e. of voltages have been reported in the same table after they have converged to an accuracy 10^{-7} after 18 iterations.

Monte Carlo Method

A computer programme for the solution of linear equations by Monte Carlo method has been written as discussed in reference (33) The values of the unknowns were computed after the play of 10 games and the values are reported in table 2-3-2.

To study the variation of the unknowns with number of games the curves are shown in figure 2-2 and the remarks as regards coloseness of the results have been also made in Table 2-3-2. A discussion of the variance reduction techniques is included in Appendix H.

Case of Unsymmetrical Matrix

A problem not very common to the power system network is that which gives the matrix A , in the set of linear equations of AX = Y form, as unsymmetrical. A set of equations of reference (27) are solved by Iterative method and Monte Carlo method and results are reported in Table 2-3-3 and 2-3-4 respectively. The equation in matrix form can be written as :

$$\begin{vmatrix} 0 \\ 1 \\ 0 \end{vmatrix} = \begin{vmatrix} 0.5 & -0.1 & -0.2 \\ -0.2 & 0.7 & -0.1 \\ +0.1 & -0.2 & 0.6 \end{vmatrix} \quad \begin{vmatrix} X_1 \\ X_2 \\ X_3 \end{vmatrix}$$

Such a situation could however arise in sequence network equation where the mutual sequence impedances may not be the same.

One thing of some remark is that while the values of X_2 and X_3 hunt around the actual value swiftly but the value of X_1 does not at all approach in the close vicinity of its actual value as shown in figure 2-3

2-4 Advantages and Limitations of the Methods discussed

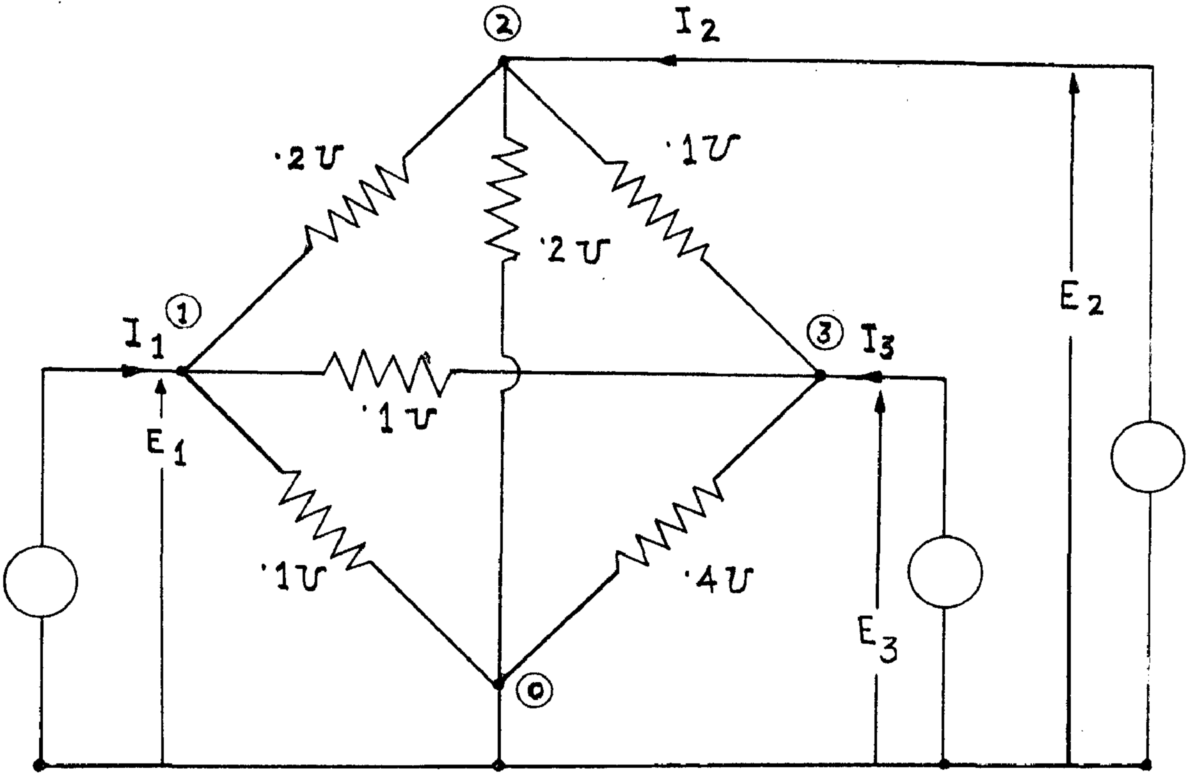
The Monte Carlo method may not give very highly accurate values as may be available by other methods such as direct,Iterative methods. However, for approximate solution the method is quite good.

Also for a large value n i.e.,the number of unknowns the other methods may not be so helpful because of their inherent time consuming nature and of their own requirements such as requiring large memory thereby reducing considerably the size of problem that can be handled.

In the Iterative method the values may or may not converge in case of large value of n, in that case it may not be of any use. Some iterations are always necessary if the values are desired to be more accurate if solution is obtained by Monte Carlo method. Reference (24,26) have suggested Newton's Approximation formula to be used to improve upon the results obtained by Monte Carlo method. But it would again cause much difficulty with large value of n. Possibly several iterations may necessary to bring the values to desired precision. However, if the Iterative method is used in conjunction with Monte Carlo Method, the following advantages are quite apparent.

1. Monte Carlo Method would atleast give approximately accurate values much faster then any other method
2. The approximately accurate values if used as the first guess values in Iterative method then there would not be much chance of the Iterative process getting diverged and not providing any solution at all.

3. Very few iterations could be necessary to improve the results to desired accuracy as is obvious from Table 2-3-5. Although for a small problem of the size considered herein the effect may not be so pronounced but the large sized problems would definitely have an advantage.



A D.C. NETWORK

FIGURE 2-1

TABLE 2-3-1

RESULTS OF ITERATIVE METHOD (SYMMETRIC MATRIX)

Accuracy taken is 10^{-7}

ITERATION NO	ERROR AFTER ITERATION
1	0.575×10^{1}
2	0.22604167×10^{1}
3	0.9217880
4	0.30652810
5	0.10476740
6	$0.356659 \times 1\bar{0}^{1}$
7	0.121485×10^{-1}
8	$0.413\ 75 \times 10^{-2}$
9	0.14091×10^{-2}
10	0.4803×10^{-3}
11	0.1632×10^{-3}
12	0.559×10^{-4}
13	0.194×10^{-4}
14	0.59×10^{-5}
15	0.22×10^{-5}
16	$0.11 \times 1\bar{0}^{5}$
17	$0.4 \times 1\bar{0}^{6}$
18	10^{-7}

VALUES AFTER 18 ITERATIONS

E_1 = 4.33734900 volts

E_2 = 2.28915640 volts

E_3 = 2.77108420 volts

TABLE 2.5.2

RESULTS OF MONTE CARLO METHOD (SYMMETRIC MATRIX)

GAME NO	VOLTAGES COMPUTED			Remarks
	E_1	E_2	E_3	
10	4.4991300	2.499550	2.749685	E_3 close to actual value
20	4.0825883	2.49955	2.3747825	
30	4.0825883	2.3329433	2.49975	E_2 close to actual value
40	3.62437	2.4370912	2.62473	
50	3.4994	2.299622	2.749715	E_2 and E_3 close to actual value
60	3.8326533	2.3329433	2.74971	
70	4.0349792	2.2139171	2.6783035	
80	3.936795	2.4370912	2.7184743	
90	4.0548288	2.2496376	2.83303	
100	4.249235	2.299622	3.0246535	E_2 close to actual value
110	4.1583504	2.1587431	2.9314922	
120	4.249245	2.2288079	2.9580058	
130	4.499130	2.2688638	2.9804403	
140	4.2492373	2.2674853	2.9818139	
150	4.1421157	2.2329693	2.9830043	
160	4.2825563	2.2964934	2.9371831	
170	4.3273403	2.3966591	2.9408588	E_1 close to actual value
180	4.2492420	2.3190636	2.9163583	
190	4.2492476	2.2627931	2.9207431	

MONTECARLO METHOD
(SYMMETRICAL)

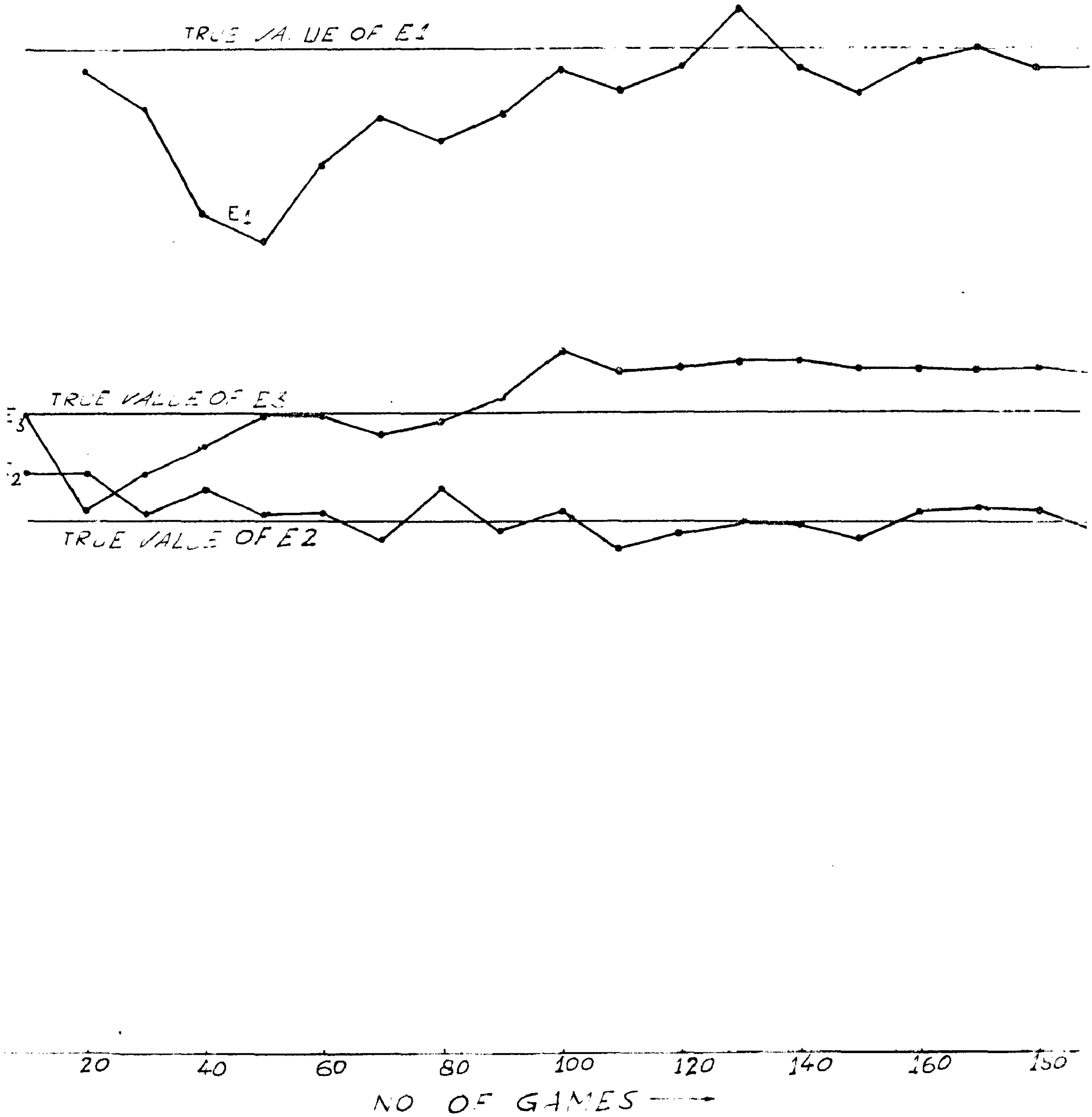


FIGURE 2-2

TABLE 2-3-3

RESULTS OF ITERATIVE METHOD (UNSYMMETRIC MATRIX)

Accuracy taken is 10^{-7}

All unknowns started with 0.0 values

ITErATION NO	ERROR AFTER ITERATION
1	0.19047618×10^{1}
2	0.82766446
3	0.18248547
4	$0.42176340 \times 10^{-1}$
5	$0.97058600 \times 10^{-2}$
6	0.223461×10^{-2}
7	0.51431×10^{-3}
8	$0.11836 \times 1\bar{0}^{3}$
9	0.2733×10^{-4}
10	0.631×10^{-5}
11	$0.134 \times 1\bar{0}^{5}$
12	0.42×10^{-6}
13	0.1×10^{-6}
14	$1\bar{0}^{7}$

VALUES AFTER 14 ITERATIONS

X_1	=	0.60606056
X_2	=	1.69696960
X_3	=	0.66666662

TABLE 2-3-4

RESULTS OF MONTE CARLO METHOD (UNSYMMETRIC MATRIX)

GAMES	UNKNOWN	COMPUTED		REMARKS
	X_1	X_2	X_3	
10	.49997	1.249925	.4999700	
20	.3749775	1.1249325	.6249625	
30	.58329833	1.249925	.666 62666	X_3 very close to actual value and X_2 close to actual value
40	.6249625	1.3124212	.56246625	
50	.549967	1.249925	.599964	
60	.58329833	1.4165816	.66662666	X_3 very close to actual value and X_2 close to actual value
70	.64281857	1.4284857	.67853071	
80	.65621062	1.4374137	.68745875	
90	.749955	1.4721338	.72217888	
100	.7249565	1.449913	.7249565	
110	.70450318	1.477184	.70450318	
120	.70829083	1.4790779	.72912291	
130	.73072538	1.4806803	.67303653	
140	.73209892	1.4820539	.64281857	
150	.73328933	1.49991	.649961	
160	.70308281	1.46 86618	.67183468	
170	.691135	1.426385	.67643	
180	.69440297	1.4582458	.66662666	X_3 very close to actual value.
190	.69732657	1.4867528	.67101236	
200	.68745875	1.5124092	.649 961	

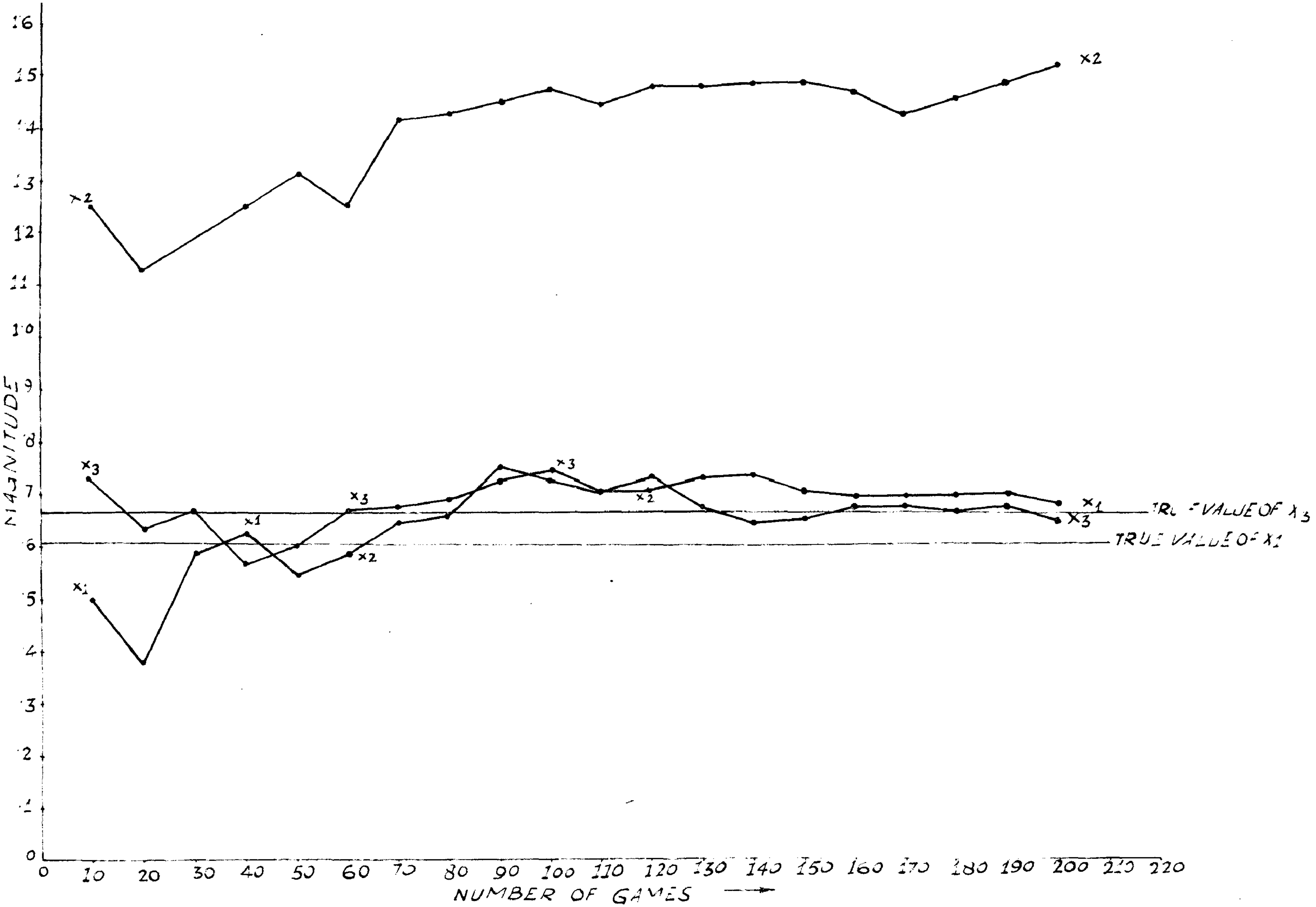

MAGNITUDE
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
0
NUMBER OF GAMES
0
10
20
30
40
50
60
70
80
90
100
110
120
130
140
150
160
170
180
190
200
210
220
x2
x3
x1
TRUE VALUE OF x3
TRUE VALUE OF x1

FIGURE 2-3

TABLE 2-3-5

ITERATIVE METHOD IN CONJUNCTION WITH MONTE CARLO

Accuracy : 10^{-7}

GUESS VALUES TAKEN AS THE VALUES OF UNKNOWNS AFTER 200 GAMES

ITERATION NO	ERROR AFTER ITERATION
1	.29920348
2	$.5616855 \times 10^{-1}$
3	$.11110169 \times 10^{-1}$
4	$.259534 \times 10^{-2}$
5	$.59694 \times 10^{-3}$
6	$.1374400 \times 10^{-3}$
7	$.3146 \times 10^{-4}$
8	$.714 \times 10^{-5}$
9	$.172 \times 10^{-5}$
10	$.52 \times 10^{-6}$
11	$.1 \times 10^{-6}$
12	10^{-7}

The values after 12 iterations are same as given in table 4-2-1

CHAPTER - 3

FIELD PLOTTING BY MONTE CARLO METHOD

Introduction

In general, a second order Partial differential equation of two independent variables x,y is of the form

$$A\ (x,y)\frac{\partial^2 y}{\partial x^2} + B(x,y)\ \frac{\partial^2 u}{\partial x \partial y} + C\ (x,y)\frac{\partial^2 u}{\partial y^2} + f(x,y,u,\frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}) = 0 \quad \ldots(1)$$

Coefficient A,B and C are generally functions of independent valuables. This equation is of course linear in the second order terms but the term $f(x,y,u, \frac{\partial u}{x}, \frac{\partial u}{y})$ may be linear or non-linear and the equation may be linear or Quasi-Linear accordingly. Aslo if $B^2 - 4AC < 0$, the equation is called Elliptical Equation. For $B^2 - 4AC = 0$, the equation is known as Parabolic Equation and finally if $B^2 - 4AC > 0$, the equation becomes Hyperbolic Equation. The present discussion concerns with the equation of type first, i.e. the Elliptical Partial differential equation.

The well known Laplace and Poisson's equations viz,

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0 \text{ and } \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(\ x,y) \quad \ldots.\ (2)$$

belongs to this catagory.

The boundary conditions of this type of equation specify either the function u or its normal derivative or the linear combination of the two at every point of the closed boundary of the region defined with in which the solution u(x,y) is desired.

To an Electrical Engineer, both the equations viz, Laplace and Poisson's Equations are of great importance. One may come across such type of equations with specified boundary conditions either for plotting of a field pattern, be it a case of machineor a high voltage equipment or electron tube or electrostatic potential in a uniform dielectric or in two dimensional condenser. The Laplace Equation is used fo compute the field values within a defined boundary and hence the solution of this type has been discussed herein by taking a irregular boundary in two-dimensional region.

The techniques for the solution of the said equation are similar for multidimentsional problem and the same method can be extended to solve them.

3.2 Different methods for solving Elliptical Equations

The Elliptical partial differential equation, in general, are solved numerically by reducing the problem to the solution of a set of simultaneous linear algebraic equations by finite difference techniques. It can be easily shown that finite-difference form for Laplace Equation is given by

$$u_{xx} + u_{yy} \approx (u_1+u_2+u_3+u_4 - 4u_0)/ h^2 = 0 \quad \ldots(3)$$

Or $u_0 = \frac{1}{4} (u_1 + u_2 + u_3 + u_4) = \frac{1}{4} \sum_{i=1}^{4} ui$

where u_1, u_2, u_3, u_4 are four neighbouring lattice points of mesh. Similarly the finite difference form of Poisson's Equation reduces to

$$u_0 = \frac{1}{4} (u_1 + u_2 + u_3 + u_4) + \frac{1}{4} h^2 f_0 \quad \ldots (4)$$

where $f0 = f (x_0, y_0)$

There are two effective methods of solving numerically the Laplace and Poissons equations, viz, Iterative method and Relaxation method.

In the former method the estimates of unknowns are improved in sequence with some initial guess at the start till the unknowns are determined to a desired degree of accuracy.

In the latter the attention is concentrated on the unknowns which seem to have greatest errors. The method assumes the mesh lineas analogous to elastic strings which are relaxed in turn during successive steps of the iteration. The starting point in any method is the same i.e. to divide the region in the form of meshes and then the values are computed at each junction point of the mesh. A Monte Carlo method is herein discussed for the solution of Laplace Equation and a comparison has been made with the values obtained by Iterative method at an interval of 50 games. The Monte Carlo method is therefore discussed in detail.

3-2 Monte Carlo Method for Laplace Equation

The following are the steps necessary to solve the equation by Monte Carlo method.

1. The region enclosed within the boundary is replaced by a rectangular mesh of lattice points. The bounding curves is replaced by a set of lattice points which are nearest neighbours to mesh points within the boundary. The meshes are generally rectangular but ~~the~~ to suite the boundary they may be taken as triangular or hexagonal.

Here only rectangular meshes would be considered.

2. To find the solution at any point P with in boundary curve C, a set of random walks of ~~frixxx~~ fittious particle is made, starting at P and terminating at the boundary points which are already specified and this finishes a game. The score of the game is the value given at the point on boundary at which the walks in that game terminate. Such several games are played and an average value per game basis gives the solution at the point P.

The part of the (x,y) plane bounded by a contour C on which is imposed a rectangular mesh with interval lengths h_1 and h_2 in the x and y directions respectively is considered and a particle performs a random walk on the mesh points subject to the following rules.

1. The probabilities of passing from (x_i, y_j) to $(x_{i-1} y_j)$, $(x_i +1, y_j)$, (x_i, y_{j-1}) and (x_i, y_{j+1}) are p_1, p_2, p_3 and p_4 respectively.

2. $\sum_i p_i = 1$, so that no other steps are possible

3. The precoess starts at an arbitrary mesh point P (x_p, y_q) and is terminated when a particle reaches boundary C, whereupon a score S, the value of which will depend upon the point at which C was reached, is associated with (x_p, y_q). This provides the boundary condition; the score S is the value taken by the solution of the differential equation at that point on the boundary.

It is clear that the average score

$S_{pq} = S\,(x_p,\, y_q)$ associated with $(x_p,\, y_q)$

satisfies the equation

$$S_{pq} - p_1\, S_{p+1},\, q - p_2\, S_{p-1,q} - p_3\, S_{p,\, q+1} - p_4\, S_{p,q-1} = 0 \qquad \ldots(5)$$

If the con itions such as

$$p_1 = p_2 = \frac{h_2^2}{2(h_1^2 + h_2^2\ )}$$

and $p_3 = p_4 = \dfrac{h_1^2}{2(h_1^2 + h_2^2)}$ are imposed, then equation

(3) becomes

$$\frac{1}{h_1^2}\,(2\,S_{pq} - S_{p+1,q} - S_{p-1,q}) + \frac{1}{h_2^2}\,(2S_{pq} - S_{p,q+1} - S_{p,q-1}) = 0 \qquad \ldots(6)$$

The equation (3) is the finite difference form of the Laplace equation in two variables.

Further this equation would reduce to

$$\frac{1}{h^2}\,(\,4S_{pq}\ \cancel{4/S_{pq}} - S_{p+1,q} - S_{p-1,q} - S_{p,q+1} - S_{p,q-1}) = 0 \quad ..(7)$$

For $h_1 = h_2$, i.e., when intervals along X and Y ax~~o~~es are same

3.3 Field Plotting By Mo~~u~~nte Carlo Method

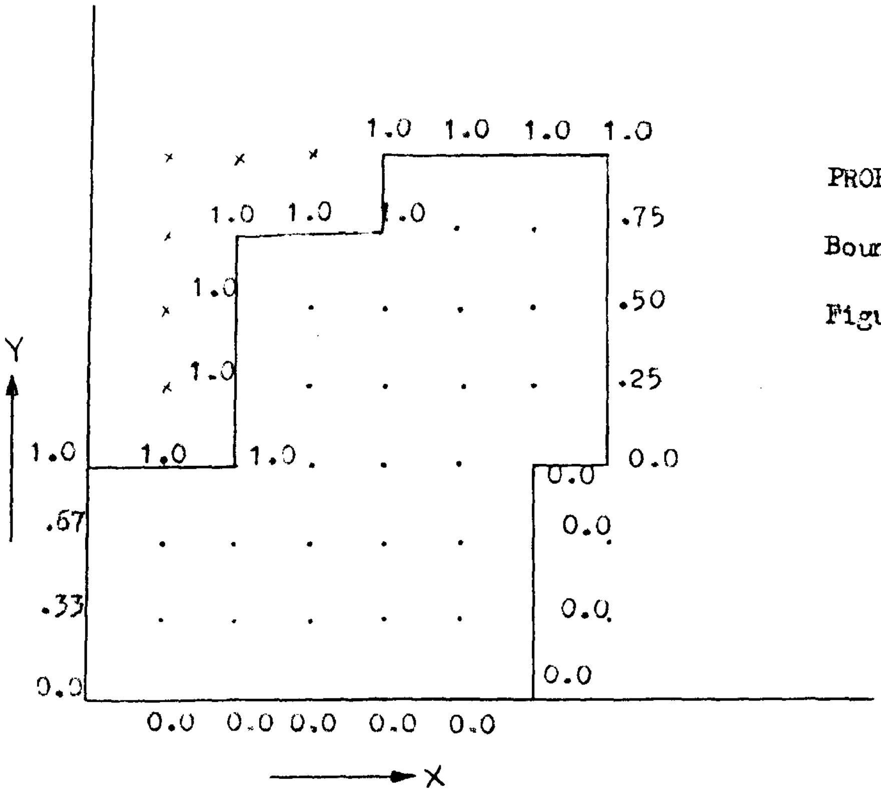
A problem (27) to illustrate the Monte Carlo method for field plotting or for the solution of Laplace Equation with the boundary as given in fig. 3.1 has been solved completely. The computer program has been written for the same and the subroutine used (Appendix D) for the generation of random numbers was taken to be the same as that used in chapter 2 for the solution of Linear Equations by Monte Carlo Method. The movementof walks was guided by the random sequence in which case the sequence has been made to contain only 1,2,3 and 4 numbers only, rest being discarded and 1 stands for the movement by one unit to the right, 2 for the upward movement, 3 for the left and 4, finally, for the movement downwards.

The solution has been obtained from 50, 100, 150 and 200 games and then average being compared with the values obtained by Iterative method in Table 3-5, The boundary values and the whole mesh network was fed in as input to the computer in the form of a matrix as shown in Appendix-D The computer programme is included in the Appendix-C and the results have been listed in Table 3-1, 3-2, 3-3 and 3-4 for 50, 100, 150 and 200 games respectively.
For comparison the values from table 3-1, 3-2, 3-3 and 3-4 have been rounded off for the inclusion in Table 3-5.
The maximum error obtained was .032 and the minimum as zero.

3-4 Merits and Demerits of Monte Carlo Method

The Monte Carlo method provides an easy method for the solution of Laplace equation even with irregular boundary and specially with multidimensional region it supercedes the conventional methods which become cumbersome and sometimes almost impossible. Field values at any point can be independently found out without finding the solution simultaneously at all the lattice points, which is essential with conventional method. The solution with Iterative method may or may not converge in certain cases but Monte Carlo atleast provides some solution to that effect. Moreover the memory space requirements are much less because the values at the four neighbouring point may not be stored at all, only the boundary points are needed to be stored. The only draw back with Monte Carlo method seems to be with the large number of games required for more accurate results.

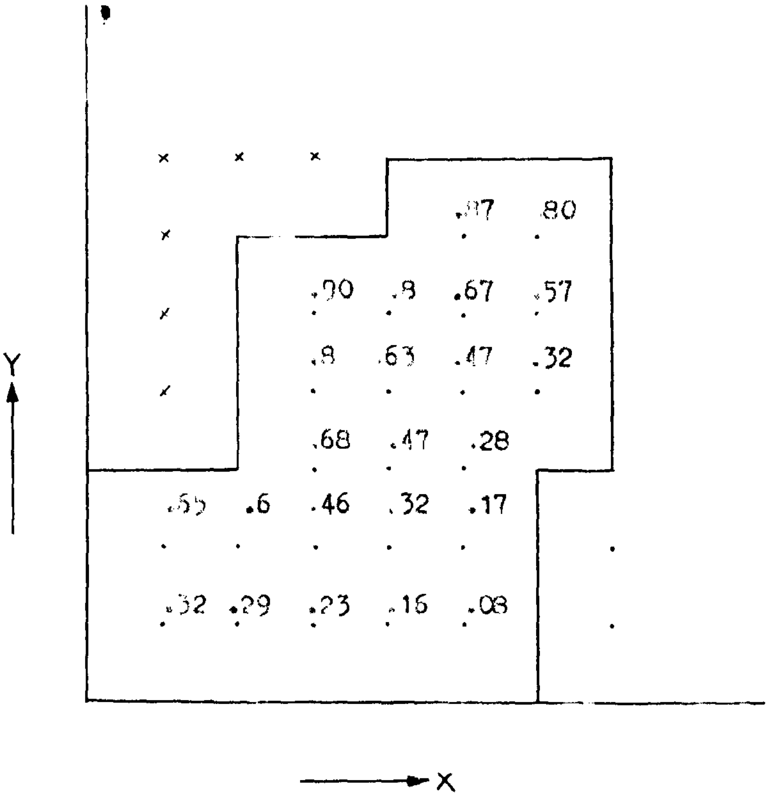
Variance may be calculted at the end of each set if game or games and the variance reducing techniques discussed in chapter 2 may be applied so as to reduce the number of trials. An agreement, generally is,necessary between the number of games and machine time. When assessing variance reducing techniques, any increase in computer time per trial which may result from their introduction must be considered.



PROBLEM OF FIELD PLOTTING

Boundary Values Shown

Figure 3-1



MONTE CARLO METHOD

FOR

FIELD PLOTTING

Actual Values

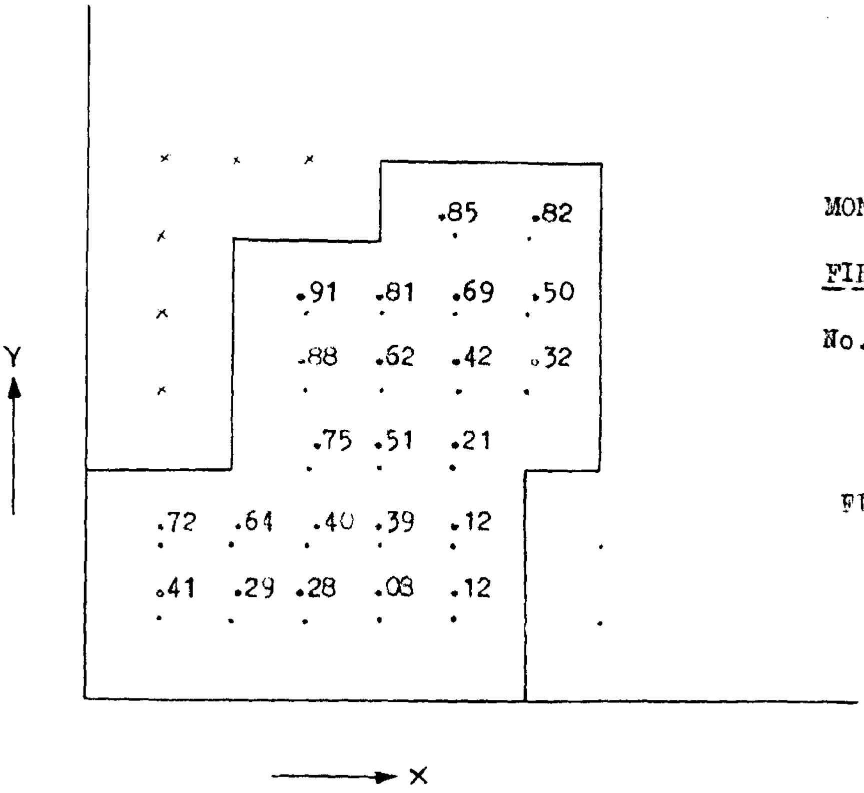
FIGURE 3-2

```
TABLE NO. 3-1

RESULTS OF THE SOLUTION  OF LAPLACE EQUATION


ANU(  2,  6)=  8.5000080E-01
ANU(  2,  7)=  8.2000020E-01
ANU(  3,  4)=  9.1500080E-01
ANU(  3,  5)=  8.1840140E-01
ANU(  3,  6)=  6.9000160E-01
ANU(  3,  7)=  4.9500200E-01
ANU(  4,  4)=  8.8000100E-01
ANU(  4,  5)=  6.2500320E-01
ANU(  4,  6)=  4.2500480E-01
ANU(  4,  7)=  3.2500420E-01
ANU(  5,  4)=  7.5500220E-01
ANU(  5,  5)=  5.1000460E-01
ANU(  5,  6)=  2.1000760E-01
ANU(  6,  2)=  7.2080120E-01
ANU(  6,  3)=  6.4680280E-01
ANU(  6,  4)=  3.9500560E-01
ANU(  6,  5)=  3.9000560E-01
ANU(  6,  6)=  1.1660840E-01
ANU(  7,  2)=  4.1260360E-01
ANU(  7,  3)=  2.8620620E-01
ANU(  7,  4)=  2.8160700E-01
ANU(  7,  5)=  8.0009000E-02
ANU(  7,  6)=  1.2000880E-01
```

TABLE NO. 3-2

RESULTS OF FIELD PLOTTING BY MONTE CARLO

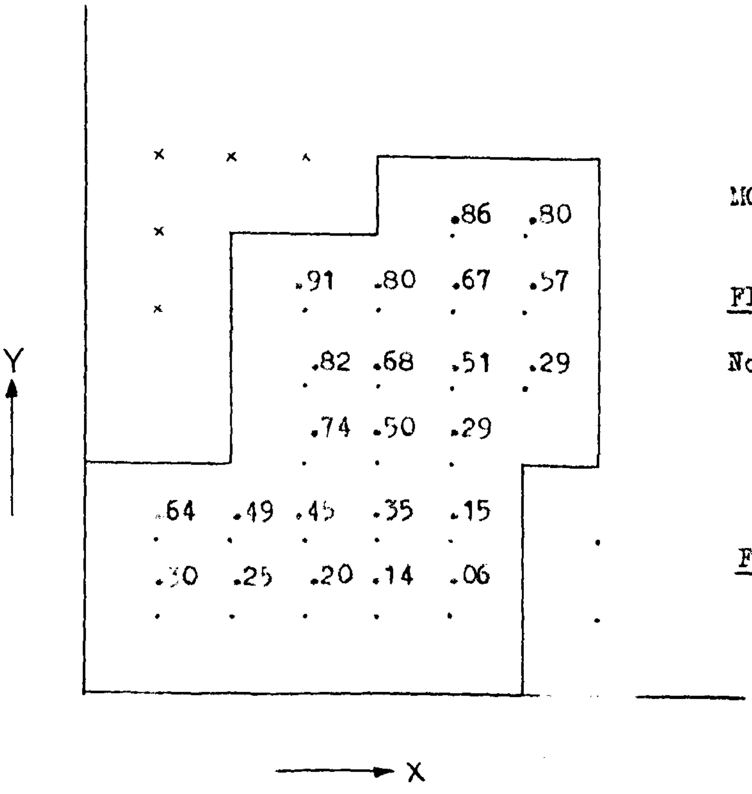
```
ANU( 2, 6)= 8.6000060E-01
ANU( 2, 7)= 8.0250080E-01
ANU( 3, 4)= 9.1170070E-01
ANU( 3, 5)= 8.0500160E-01
ANU( 3, 6)= 6.7500230E-01
ANU( 3, 7)= 5.7250170E-01
ANU( 4, 4)= 8.2000170E-01
ANU( 4, 5)= 6.8590250E-01
ANU( 4, 6)= 5.1750350E-01
ANU( 4, 7)= 2.9170430E-01
ANU( 5, 4)= 7.4830230E-01
ANU( 5, 5)= 5.0590450E-01
ANU( 5, 6)= 2.9000670E-01
ANU( 6, 2)= 6.3740210E-01
ANU( 6, 3)= 4.8740430E-01
ANU( 6, 4)= 4.4520510E-01
ANU( 6, 5)= 3.5420640E-01
ANU( 6, 6)= 1.4920840E-01
ANU( 7, 2)= 3.0240460E-01
ANU( 7, 3)= 2.5330690E-01
ANU( 7, 4)= 2.0350760E-01
ANU( 7, 5)= 1.3830840E-01
ANU( 7, 6)= 6.0009400E-02
```



MONTE CARLO METHOD
FOR
FIELD PLOTTING

No. OF GAMES 50

FIGURE 3-3



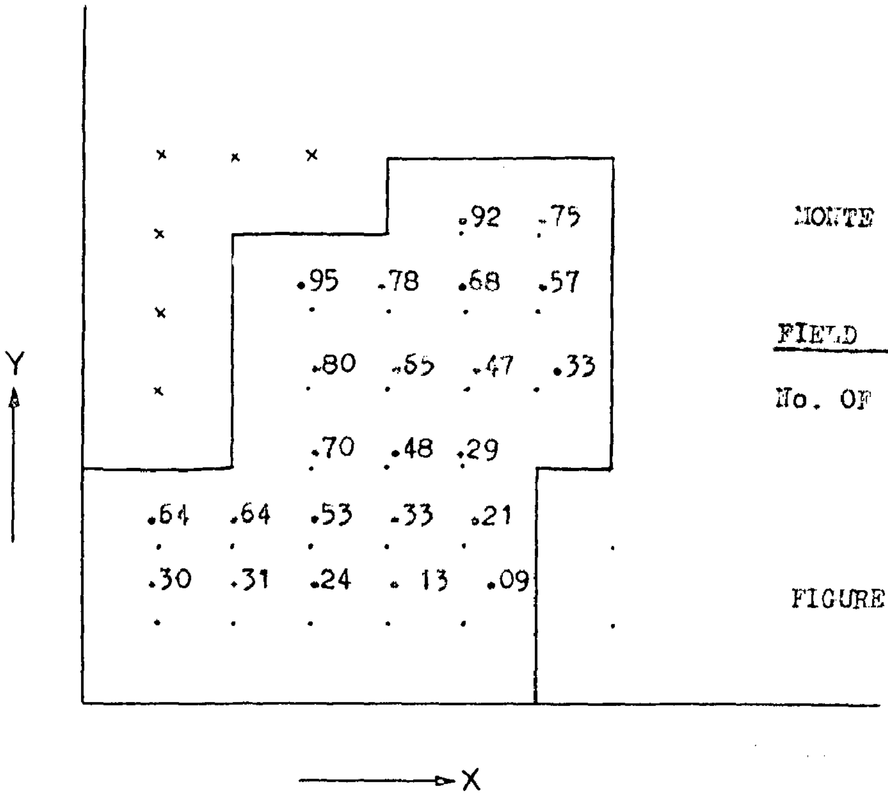
MONTE CARLO METHOD
FOR
FIELD PLOTTING

No. OF GAMES 100.

FIGURE 3-4

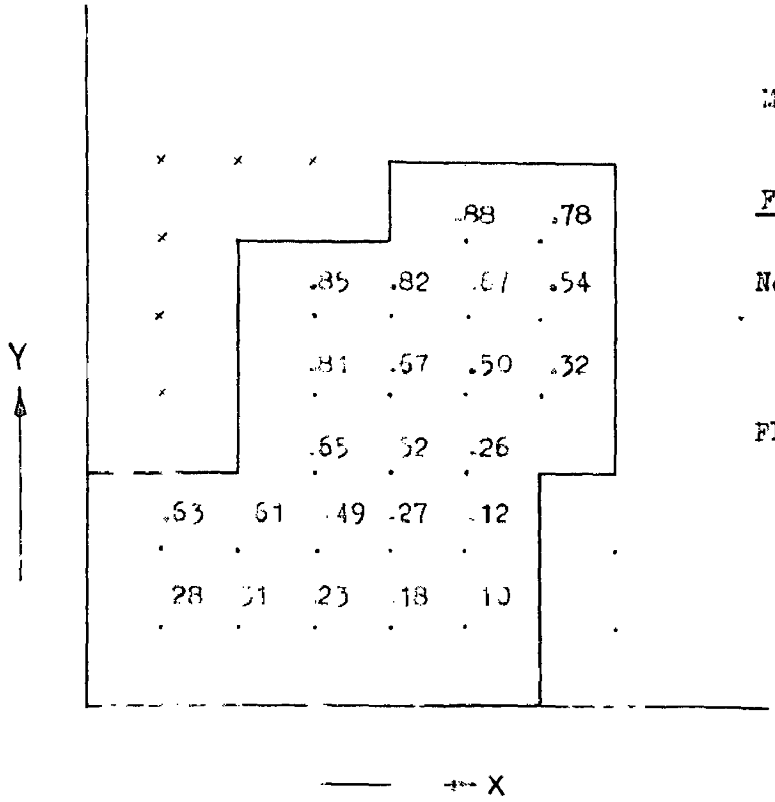
```
         TABLE NO. 3-3


         RESULTS OF THE SOLUTION  OF LAPLACE EQUATION


ANU(  2,  6)=  8.8833380E-01
ANU(  2,  7)=  7.8166733E-01
ANU(  3,  4)=  8.5280126E-01
ANU(  3,  5)=  8.2946806E-01
ANU(  3,  6)=  6.7666846E-01
ANU(  3,  7)=  5.4000186E-01
ANU(  4,  4)=  8.1220153E-01
ANU(  4,  5)=  6.7113573E-01
ANU(  4,  6)=  5.0167073E-01
ANU(  4,  7)=  3.2000413E-01
ANU(  5,  4)=  6.5007900E-01
ANU(  5,  5)=  5.2167080E-01
ANU(  5,  6)=  2.6114033E-01
ANU(  6,  2)=  6.3820200E-01
ANU(  6,  3)=  6.1126003E-01
ANU(  6,  4)=  4.0000473E-01
ANU(  6,  5)=  2.7454040E-01
ANU(  6,  6)=  1.1824186E-01
ANU(  7,  2)=  2.7913813E-01
ANU(  7,  3)=  3.1333053E-01
ANU(  7,  4)=  2.3734066E-01
ANU(  7,  5)=  1.8000813E-01
ANU(  7,  6)=  1.0107546E-01
```

```
        TABLE NO. 3-4

        RESULTS OF THE SOLUTION  OF LAPLACE EQUATION


ANU(  2,  6)=  9.2125020E-01
ANU(  2,  7)=  7.5875000E-01
ANU(  3,  4)=  9.2600055E-01
ANU(  3,  5)=  7.8750160E-01
ANU(  3,  6)=  6.8200190E-01
ANU(  3,  7)=  5.7250120E-01
ANU(  4,  4)=  8.0955160E-01
ANU(  4,  5)=  6.5125310E-01
ANU(  4,  6)=  4.4625415E-01
ANU(  4,  7)=  3.3000355E-01
ANU(  5,  4)=  7.0415255E-01
ANU(  5,  5)=  4.8335480E-01
ANU(  5,  6)=  2.8500665E-01
ANU(  6,  2)=  6.4405195E-01
ANU(  6,  3)=  6.4805305E-01
ANU(  6,  4)=  5.3255440E-01
ANU(  6,  5)=  3.3625630E-01
ANU(  6,  6)=  2.1165765E-01
ANU(  7,  2)=  2.0925470E-01
ANU(  7,  3)=  3.1320610E-01
ANU(  7,  4)=  2.4005745E-01
ANU(  7,  5)=  1.3165860E-01
ANU(  7,  6)=  9.5808950E-02
```



MONTE CARLO METHOD
FOR
FIELD PLOTTING
No. OF GAMES 200

FIGURE 3-6



MONTE CARLO METHOD
FOR
FIELD PLOTTING

No. OF GAMES 150

FIGURE 3-5

TABLE No. 3-5

POINT	Values for 50 Games	Values for 100 Games	Values for 150 Games	Values for 200 Games	Average Value	Actual Value	Difference Actual∼ Average
(2,6)	.85	.86	.88	.92	.875	.87	.005
(2,7)	.82	.80	.78	.75	.79	.80	.010
(3,4)	.91	.91	.85	.95	.905	.90	.005
(3,5)	.81	.80	.82	.78	.802	.80	.002
(3,6)	.69	.67	.67	.68	.677	.67	.007
(3,7)	.50	.57	.54	.575	.545	.57	.025
(4,4)	.88	.82	.81	.80	.827	.80	.027
(4,5)	.62	.68	.67	.65	.655	.63	.025
(4,6)	.42	.51	.50	.47	.475	.47	.005
(4,7)	.32	.29	.32	.33	.31	.32	.010
(5,4)	.75	.74	.65	.70	.706	.68	.026
(5,5)	.51	.50	.52	.48	.502	.47	.032
(5,6)	.21	.29	.26	.28	.26	.28	.020
(6,2)	.72	.64	.63	.64	.657	.65	.007
(6,3)	.64	.49	.61	.64	.595	.60	.005
(6,4)	.40	.45	.49	.53	.467	.46	.007
(6,5)	.39	.35	.27	.33	.335	.32	.015
(6,6)	.12	.15	.12	.21	.15	.17	.020
(7,2)	.41	.30	.28	.30	.322	.32	.002
(7,3)	.29	.25	.31	.31	.29	.29	.000
(7,4)	.28	.20	.23	.24	.237	.23	.007
(7,5)	.08	.14	.18	.13	.132	.16	.028
(7,6)	.12	.06	.10	.09	.092	.08	.012

CHAPTER - 4

SETTING UP OF SELF AND MUTUAL IMPEDANCES ON DIGITAL COMPUTER

4-1 Definition

According to American standard Association the driving point impedance at any pair of terminals of a network is defined as the ratio of an applied potential difference to the resulting current at these terminals, all other terminals being terminated in a specified manner.

Similarly the transfer impedance is the ratio of a potential difference applied at one pair of terminals to the resultant current at the other pair of terminals, all terminals being terminated in a specified manner. When the other terminal pairs are open circuited the Driving and Transfer impedances thus evaluated are termed as Self and Mutual impedances.

Importance and Uses

There are large number of Power System problems in which the driving point and Transfer impedances of the network are required. They , as a circuit analysis technique, have been extensively used in load flow, short circuit, regulation, stability and transmission loss studies.

The driving point and transfer impedances as a practice, are found by measurements on A.C. Network analyser, which requires setting up the whole network on Analyser and by adjustments and measurements the impeaances are obtained.

Again if the impedances thus found are to be used for further computations on Digital Computer they have to be transferred to punch cards which may involve personal error and inaccurate values may be transferred.

Digital method of finding these processes ~~has~~ an advantage over the analogue method in that it gives highly accurate values which are necessary in the determination of open circuit impedances or self and mutual impedances for the use in loss formula. Moveiover, they may economically be obtained on Digital computer with minimum set up time. This also saves time when they are to be used for further computations and are required to be transferred on to ~~such~~ cards.

4-2 <u>Different Methods</u>

A few methods are suggested in reference (35) and then by Ward and Hale (37).

The method suggested by Ward and Hale (37) is based on that given by R.Bruce Shipley (36) as to the method of approach only without giving details of programming.

The methods prevalent are:

1. Matrix Method
2. Calculation by Impressed currents
3. Iterative method.

Here the Iterative method has been discussed at par and the computer programme on IBM 1620 has been drawn incorporating some special features to be discussed later.

Three systems are studied and the results of Iterative process have been analysed in details.

Matrix Method

The first step is to set up the branch impedance matrix Z branch inaccordance with the equation,

$$E \text{ branch} = Z_{\text{branch}} I_{\text{branch}} \quad .... \qquad (1)$$

Z_{branch} matrix consists of the line and transformer impedances of the network as its elements. The configuration of the network is defined by means of transformation matrix C_t, which relates the voltage at each bus to the voltage across each of branch elements and also the voltage acting around each loop in the network to the branch voltage, i.e.,

$$E_{\text{bus, loop}} = C_t \; E \text{ Branch} \quad .... \qquad (2)$$

The loop current and branch current should, for positive direction, flow in the same direction. The matrix C_t should also take into account the effect of off nominal turns ratio of the transformer.

Now using Kron's analysis further development can be carried out as follows:

If voltages and currents are related in a circuit by

$$E_{\text{old}} = Z_{\text{old}} I_{\text{old}} \quad \text{------} \qquad (3)$$

The new set of voltages could be related to old set by

$$E_{\text{new}} = C_t{}^{*} E_{\text{old}} \qquad (4)$$

then for the impedances to be transformed such that input power is invariant, there are,

$$I_{\text{old}} = C \; I_{\text{new}} \qquad (5)$$

$$Z_{\text{new}} = Ct^{*} Z_{\text{old}} C \qquad (6)$$

where C is obtained by interchanging the rows and columns of ct. Ct* being obtained by conjugating the elements of Ct.

Also $E_{new} = Z_{new} I_{new}$ (7)

Thus from above relations, one gets

$I_{branch} = C\ I_{bus,loop}$ (8)

$Z_{bus,loop} = Ct^{*}Z_{branch}C$ (9)

Also

$E_{bus,loop} = Z_{bus,loop}\ I_{bus,loop}$ (10)

The impedance matrix $Z_{bus,loop}$ could be compounded as as follows

$$\begin{bmatrix} E_{bus} \\ E_{loop} \end{bmatrix} = \begin{bmatrix} Z_1 & Z_2 \\ Z_3 & Z_4 \end{bmatrix} \begin{bmatrix} I_{bus} \\ I_{loop} \end{bmatrix} \qquad (11)$$

but as $E_{loop} = 0 = Z_3 I_{bus} + Z_4\ I_{loop}$

$Z_4 I_{loop} = - Z_3 I_{bus}$

By pre-multiplying through by the inverse of Z_4 the following can be obtained.

$I_{loop} = -Z_4^{-1}\ Z_3\ I_{bus}$ (12)

Substituting (12) in (11),

$E_{bus} = Z_1 - Z_2\ (z_4)^{-1} Z_3 \quad I_{bus}$ (13)

$= Z_{bus}\ \ I_{bus}$

Thus $Z_{bus} = Z_1 - Z_2\ (Z_4)^{-1}\ Z_3$ (14)

Z_4 is the network loop impedance matrix which is required to be inverted for the calculation of Z_{bus} matrix which defines the self and mutual impedances. The matrix method has the disadvantage in that it requires the inversion of a complex matrix as a subroutine programme and thus limits the size of the network.

The storage requirements are specifically more although the programming is straight forward. IBM 1620 has a subroutine for the inversion of complex matrix up to a max. size of 17 x 17 and therefore a system with more then 17 loop can not be easily handled on this computer.

Impressed Current Method

The method is same as the procedure used on A.C. Network analyser, i.e., impressing current at each bus and finding the voltage distribution, successively. The steps can be given as follows.

1. A current 1 + jo is impressed at generator or load bus in the notwork with reference bus grounded.
2. Current flow is assumed from energized bus to the reference bus.
3. The voltages in each branch is computed and that acting in each loop, by summing the branch voltages in that loop.
4. The balancing current required to make summation of the voltages around each loop equal to zero, is computed.
5. Superimposing the balancing flow, determined in step 4 on assumed flow of step 1, the branch voltages due to exact flow are determined.

The voltages thus determined are numerically equal to the self and mutual impedances since impressed current is 1 + jo. The current is then impressed thrn by turn to each bus and completed set of impedances can be known.

This method is superior to the matrix method because of its less memory requirements and less computing time and also it can handle larger system than that by matrix method.

Iterative Method

Iterative method consists of successively improving an assumed set of bus voltages until Kirchoff's Law has been satisfied with in the precision desired, at each node. The sum of all currents entering a bus p must be zero.

or

$$\sum_q Y_{pq} (E_q - E_p) = 0 \qquad (15)$$

where Y_{pq} is admittance between bus p and q and E_p and E_q are the voltages at bus p and q respectively.

It is obvious that

$$(\sum_q Y_{pq}) E_p = \sum_q Y_{pq} E_q$$

and

$$E_p = \frac{\sum_q Y_{pq} E_q}{\sum_q Y_{pq}} \qquad (16)$$

The equation (15) and (16) can be modified to take in to account the off-nominal turns ratio.

The method requires the setting up of driving bus voltage to 1 + jo and an initial voltage distribution is assumed (usually as 0 + j0) as the reference bus is set to 0 + j0. Thus the bus voltages are computed using equation (16) sequencially with the best known values of voltages of other buses. When all voltages have been calculated, the calculation is repeated with the improved set of voltages. This goes on till the voltages have converged to a desired precision. The acceleration factors are used to speed

The new values after applying acceleration factor are stored and calculated from

$$E = \text{Acce}\overset{le}{\wedge}\text{ration Factor } (E_{new} - E_{old}) + E_{old}$$

Once the voltages have conve.ged the input current at the energised bus is computed from

$$I_n = \sum_q Y_{nq} (1 \underline{*} E_q) \qquad (17)$$

This should also be modified if off nominal turns ratio is to be taken into account.

With all the voltages found and input current at the driving bus the self and mutual impedances are computed by dividing the voltages at all buses by input current. This process is repeated till all buses have been energized in turn and complete matrix of self and mutual impedances is known.

The method is again superior to the matrix method as regards time and system size.

Although this method requires elaborate programming but is superior than other methods.

The computer programme as listed in Appendix E for this method has been written and self and mutual impedances of different systems are found.

A study has been also made on the Iterative Process used in this method and observations have been included herein.

4-3. Illustrative Problems

The self and mutual impedances for the systems shown in figures 4.8, 4.2, 4-9 i.e.,for 4,5 and 6 buses systems are computed using the programme written by the author.

A four bus system is a ring system and the five and six bus systems contain two and three loops.
The self and mutual impedances thus found have been reported in Table 4-3-1.

Programme Features

The programme for IBM 1620 has been written in such a way as to allow maximum flexibility in the study of convergence and to disclose more information regarding the iterative process. The precision to which the result are desired can be varied at will and the acceleration factor also may be subjected to variation.

The voltages may be punched after they have converged to the desired precision after certain number of iterations which can also be known. Similarly the impedances can be directly punched even without knowing the information regarding voltages etc. ~~available in IBM 1620~~. Because of the limitations of memory spaces available in IBM 1620 the programme can handle a system upto 30 buses only directly and automatically. However, by breaking the programme in parts the system size can be increased still further.

4-4 Study on Iterative Process

Using the flexibility provided by the programme chalked out a study on Iterative process has been made so as to know the nature of convergence, the optimum acceleration factor and variation of number of iterations necessary with desired precision and for a particular system under study. The observations made are being reported herewith.

1. Convergence of Voltages

The voltages converge alternatively and not unidirectionally. They converge rapidly in the beginning and then slowly for further iterations.

The system for which the voltages have been studied is a 4 bus system of Fig. 4-8.

As is clear from the Table 4-4-1 and figure 4-1 the voltages have almost converged for 6 iterations but to achieve an accuracy of 10^{-8}, few more iterations are necessary. The voltage converge with a precision of 10^{-8} for 14 iterations.

2. Optimum Acceleration factor

The optimum acceleration factor is that value of acceleration factor for which the total number of iterations are minimum. The optimum acceleration factor depends more or less on the system under study. For a 5-bus system with a precision of 10^{-8} it has been found to be 1.1. The optimum acceleration factor generally lies between 1.0 to 2.0 but seems to have a maximum as 1.6 or 1.7 as reported by V.Gomeati (20) and W.F.Tinney. Reference 15) suggests an acceleration factor of 1.3 as quite promising, but in the view of the author this value is alright for the moderate sized system, since the optimum acceleration factor seems to vary with the size of the system.

Here it has been observed that for a four-bus system the value of optimum acceleration factor lies between 1.0 and 1.1 or to say 1.04 as is obvious from the figure 4-8 and Table 4-4-6. Similarly for a five bus sytem

it has been observed to have a value of 1.1 figure 4-3, Table 4-4-2 and for a six bus system the value is 1.2 as is obvious from the figure 4-10.
Reference (35) studied a system with 9 buses so the acceleration factor of 1.3 is justified but for larger system the optimum acceleration factor would be higher but usually less than 1.6 or 1.7 which is a quite good choice for sufficiently large system.

3. Number of Iterations

The nature of variation of no. of iterations for any particular bus voltage and system can be seen from the figures 4-3 and 4-5. This has been observed to depend upon the order in which the iterations have been performed during the process and also labelling of the reference bus. Different buses have been observed to have their minimum no. of iteration at different acceleration factors. It is not necessary to have the same acceleration factor as optimum for individuals and the cumulative iterations. curve fig 4-4.
The number of iterations have been found to vary linearly with the precision desired on logarithmic scale as is shown in figure 4-7 and is obvious from the tables 4-4-2, 4-4-3 and 4-4-4.

The number of total iterations for precision of 10^{-4} are found to be approximately half of that for the precision of 10^{-8}.

Hence the total number of iterations are predictable in advance for any value of precision if they are known for any one precision.
The optimum acceleration factor is thus the same for any precision desired for a system, it is almost independent of precision as is obvious from fig. 4-6. The total no. of iterations are almost the same for any order of iterative process and regard less of the reference bus us is clear from Table 4-4-5. There has been only one exception for the case when bus no. 4 was taken as reference bus otherwisethey are practically the same.

The total number of iterations for the systems under study have been observed to vary at the rate of 23 iterations approximately per increase in the no. of buses, as is shown in table 4-4-8 and figure 4-11. They were observed for the acceleration factor of 1.3 and accuracy of 10^{-8}.

In short for an accuracy of 10^{-8}, the average no. of iterations comes to 13 iterations per bus for an acceleration factor of 1.3
Generally, the accuracy of 10^{-8} is unnecessary except in certain cases such as loss formula calculations etc. With a.c.network one may be even satisfied with accuracy of 10^{-3}.

The digital computer thus has an advantage as regards accuracy.

The time per iteration has been observed in all the above studies as an average of 1.5 secs/per iteration.

4-4 Conclusions

The digital method thus is quite promising as regards time accuracy and cost in comparison to the analogue method.

Iterative method of finding self and mutual impedances, although may be somewhat difficult to programme, has its own advantages over the matrix and impressed current methods as far as time is concerned.

Memory requirements are much more in Matrix method than the other two. Matrix and Impressed currents methods require inversion of matrix of the size equal to the number of loops and are limited in this respect. For smaller systems any method may be used.

TABLE 4-3-1

SELF AND MUTUAL IMEDANCES CALCULATED BY ITERATIVE METHOD

1. FOR 4 BUS RING SYSTEM

Reference Bus is No. 3

Z_{11} = .09388496 + j .37494901

Z_{12} = .03765745 + j .15001102

Z_{13} = 0 + j0

Z_{14} = .03128257 + j .12511018

Z_{22} = .07514993 + .29989340

Z_{23} = 0 + j0.

Z_{24} = .01254754 +j .05005455

Z_{43} = 0 +j0

Z_{44} = .04383011 +j .17576474

2. FOR 5 BUS TWO LOOPS SYSTEM

Reference Bus is No. 5

Z_{11} = .03870969 + j.15483871

Z_{12} = .02258065 + j .09032256

Z_{13} = .01129032 + j .04516127

Z_{14} = .029=3226 + j .11612902

Z_{15} = 0 +j0

Z_{22} = .05483871 + j .21935482

Z_{23} = .02741935 + j.10967740

Z_{24} = .04193549 + j .16774191

Z_{25} = 0 +j0

Z_{33} = .03870969 + j .15483871

Z_{34} = .02096775 + j .08387096

Z_{35} = 0 + j0

Z_{44} = .09677421 + j .38709677

Z_{45} = 0 + j0

3. FOR 5 BUS SYSTEM

Reference Bus being 1

Z_{22} = .04838710 + j .19354841

Z_{23} = .03225806 + j .12903226

Z_{24} = .02903227 + j .11612905

Z_{25} = .01612903 + j .06451673

Z_{33} = .05483872 + j .21935484

Z_{34} = .019 35484 + .07741935

Z_{35} = .02747936 + j . 10967741

Z_{44} = .07741937 + j .30967745

Z_{45} = .00967741 + .03870965

Z_{55} = . .03870969 + .15483871

TABLE 4-3-1

3. SELF AND MUTUAL IMPEDANCES FOR 6-BUS, THREE LOOP SYSTEM

Reference Bus is 5th

Z_{11}	=	.0350937 + j .14015747
Z_{12}	=	.02322835 + j.09291336
Z_{13}	=	.01496063 + j .05984250
Z_{14}	=	.02795275 + j .011181100
Z_{15}	=	0 + j 0
Z_{16}	=	.02165355 + j.08661415
Z_{22}	=	.05472442 + j .21889764
Z_{23}	=	.02677166 + j. 10708660
Z_{24}	=	.04212600 + j .16850394
Z_{25}	=	0 + j0
Z_{26}	=	.02559055 + j .10236219
Z_{33}	= .0	.03503937 + j .14015748
Z_{34}	=	.02204724 + j.08818896
Z_{35}	=	0 + j0
Z_{36}	=	.02834645 + j .11338582
Z_{44}	=	.09645671 + j.38582678
Z_{45}	=	0 + j0
Z_{46}	=	.02401574 + j .09606295
Z_{66}	=	.05944882 + j .23779526

TABLE 4-4-1

NATURE OF COVERGENCE OF VOLTAGES : PRECISION 10^{-8}

ITERATION No:	VOLTAGE OF BUS NO.2		VOLTAGE OF BUS NO. 4	
	REAL	IMAG	REAL	IMAG
1	.520187	-.000312	.433738	.000144
2	.364130	-.000218	.3036166	.000101
3	.410947	-.000246	.342653	.000114
4	.396902	-.000238	.330942	.000110
5	.401116	-.000240	.334455	.000111
6	.399852	-.000239	.333407	.000110
7	.400231	-.000240	.333717	.000111
8	.400117	-.000200	.333622	.000111
		CONVERGED VOLTAGES		
14	.400143	-.000240	.333644	.000111

The above values are for a 4-bus System of Figure 4-8

CONVERGENCE OF VOLTAGES

A FOUR-BUS SYSTEM

BUS NO. 1 - ENERGISING BUS
&
BUS NO 3 - REFERENCE BUS

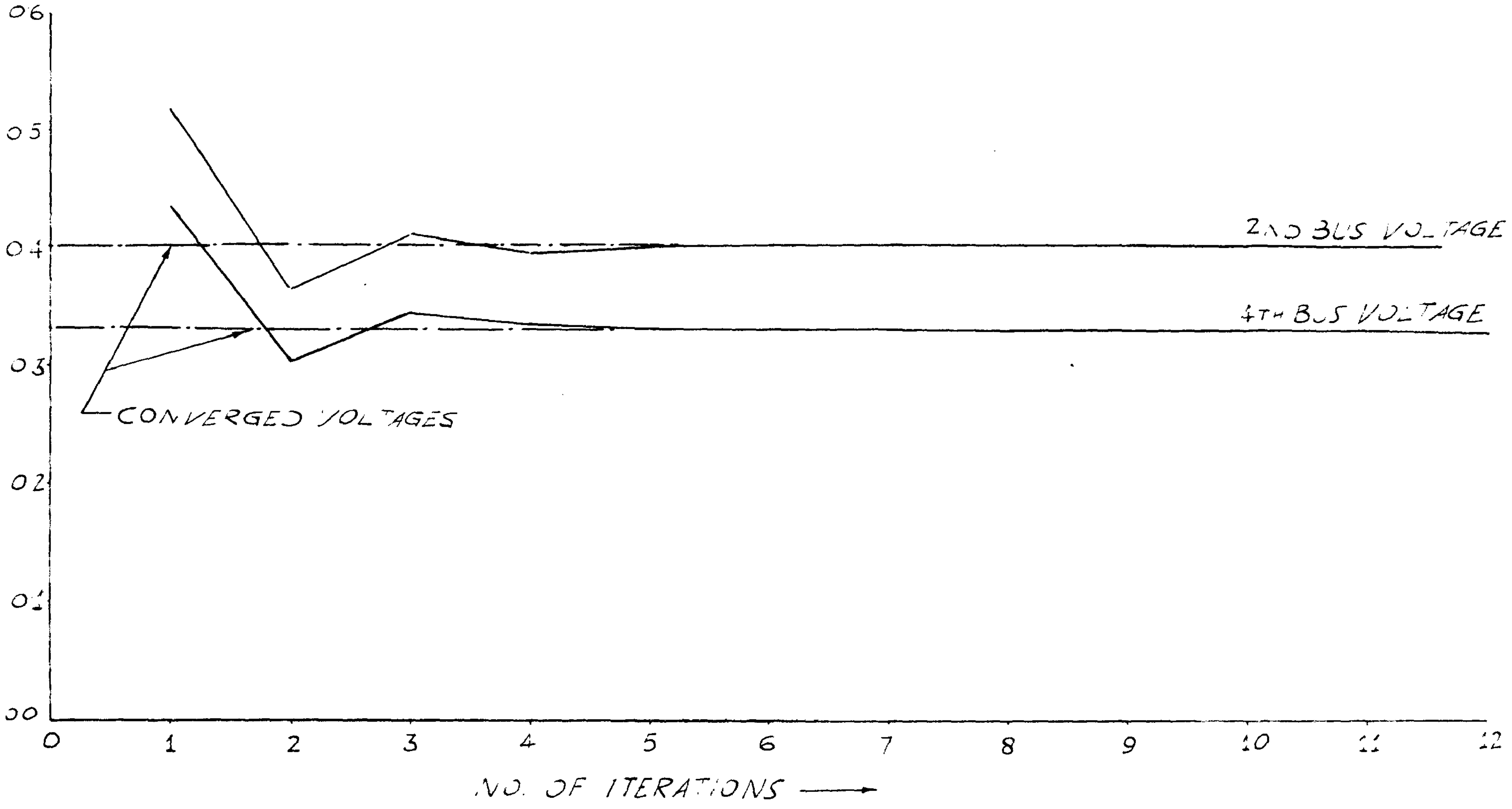
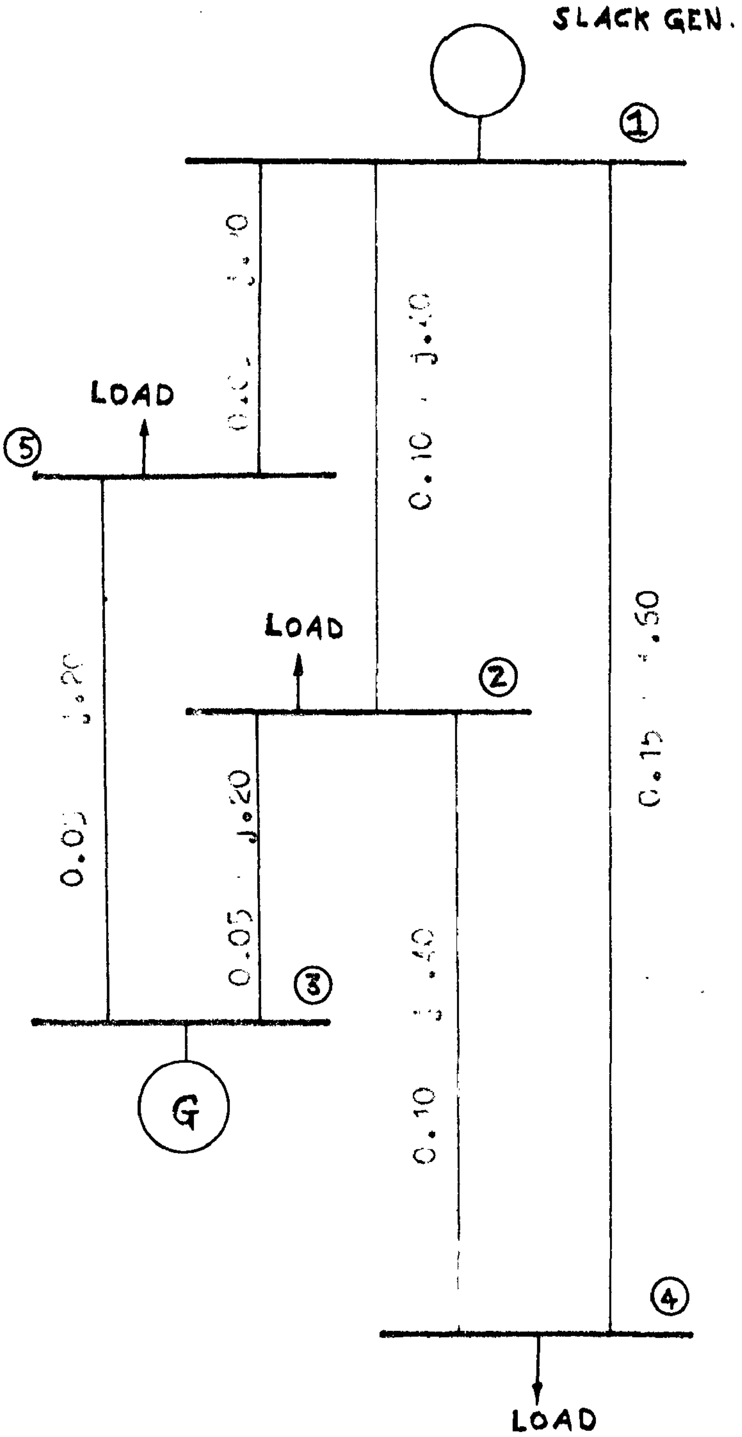


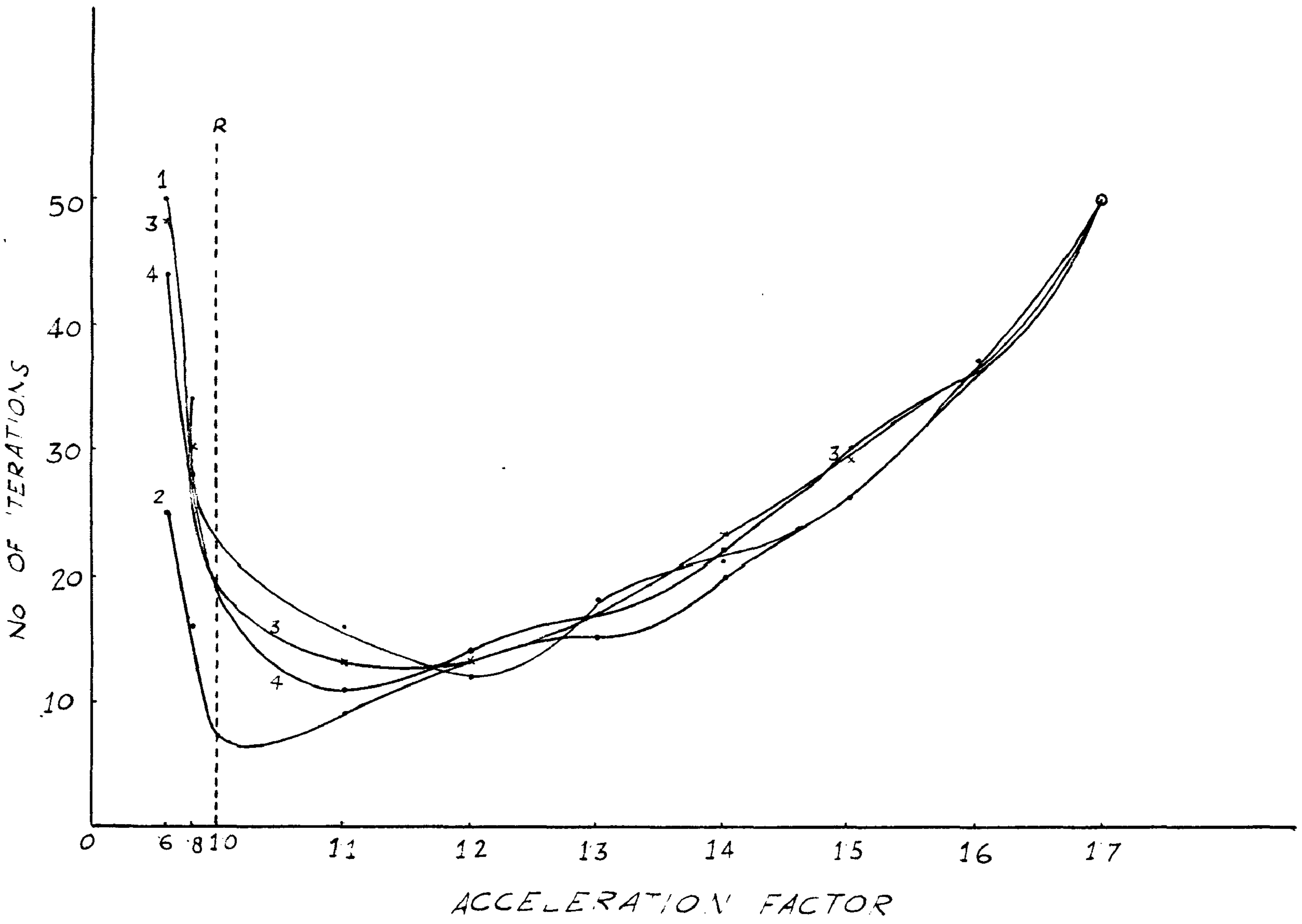
FIGURE 4-1



ONE LINE DIAGRAM

BRANCH IMPEDANCES SHOWN

FIGURE 4-2



ITERATIVE PROCESS FOR 5-BAR SYSTEM

ACCURACY 10^{-8}

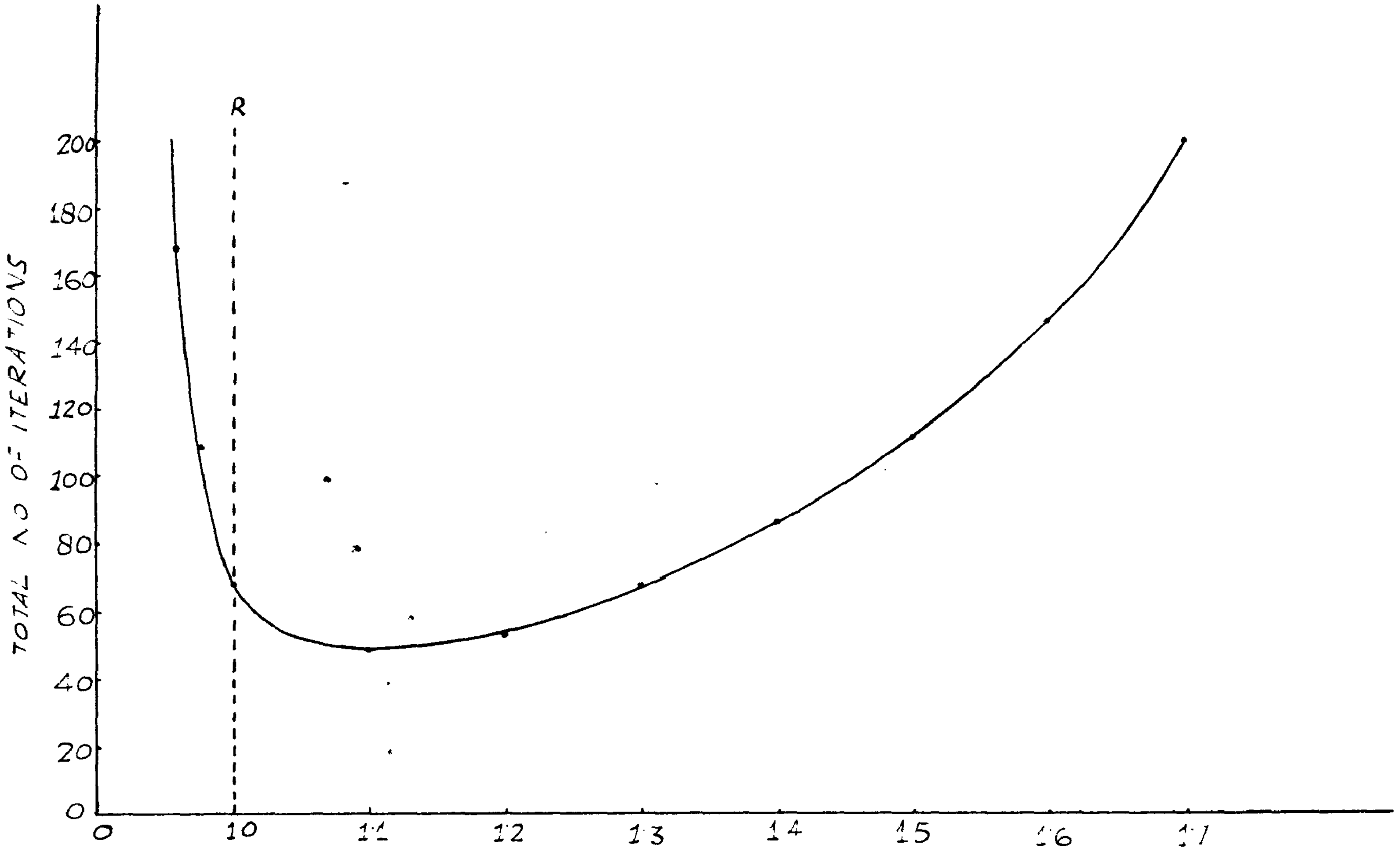
FIGURE 4-3

TABLE 4-4-2 5- BUS SYSTEM

PRECISION 10^{-8}

BUS NO	NO. OF ITERATIONS										
	Accerlation factor -	0.6	0.8	1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7
1		50	34	22	16	$\boxed{12}$	18	21	26	37	50
2		25	16	$\boxed{07}$	09	13	15	20	26	36	50
3		48	30	19	$\boxed{13}$	13	17	23	29	36	50
4		44	28	19	$\boxed{11}$	14	17	22	30	36	50
Total No. of ITERATIONS		167	108	67	$\boxed{49}$	52	67	86	111	145	200

The values in parallelogram are the minimum ones for that set.



ACCELERATION FACTOR

ITERATIVE PROCESS FOR 45 BUS SYSTEM

PRECISION FACTOR 10^{-8}.

FIGURE 4-4

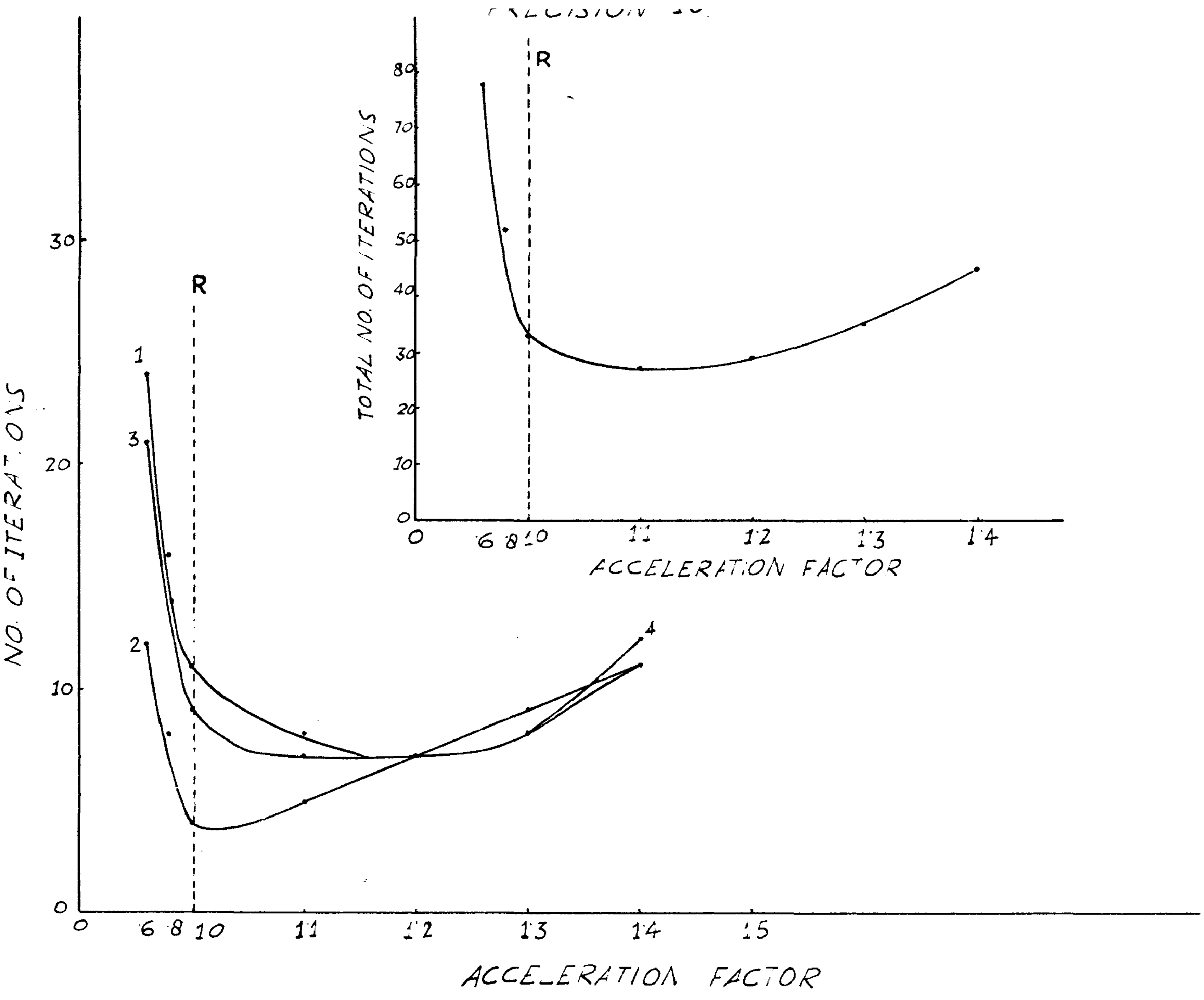

PRECISION 10
R
80
70
60
50
40
30
20
10
0
TOTAL NO. OF ITERATIONS
0
·6
·8
1·0
1·1
1·2
1·3
1·4
ACCELERATION FACTOR
R
30
20
10
0
NO. OF ITERATIONS
1
3
2
4
0
·6
·8
1·0
1·1
1·2
1·3
1·4
1·5
ACCELERATION FACTOR

FIGURE 4-5

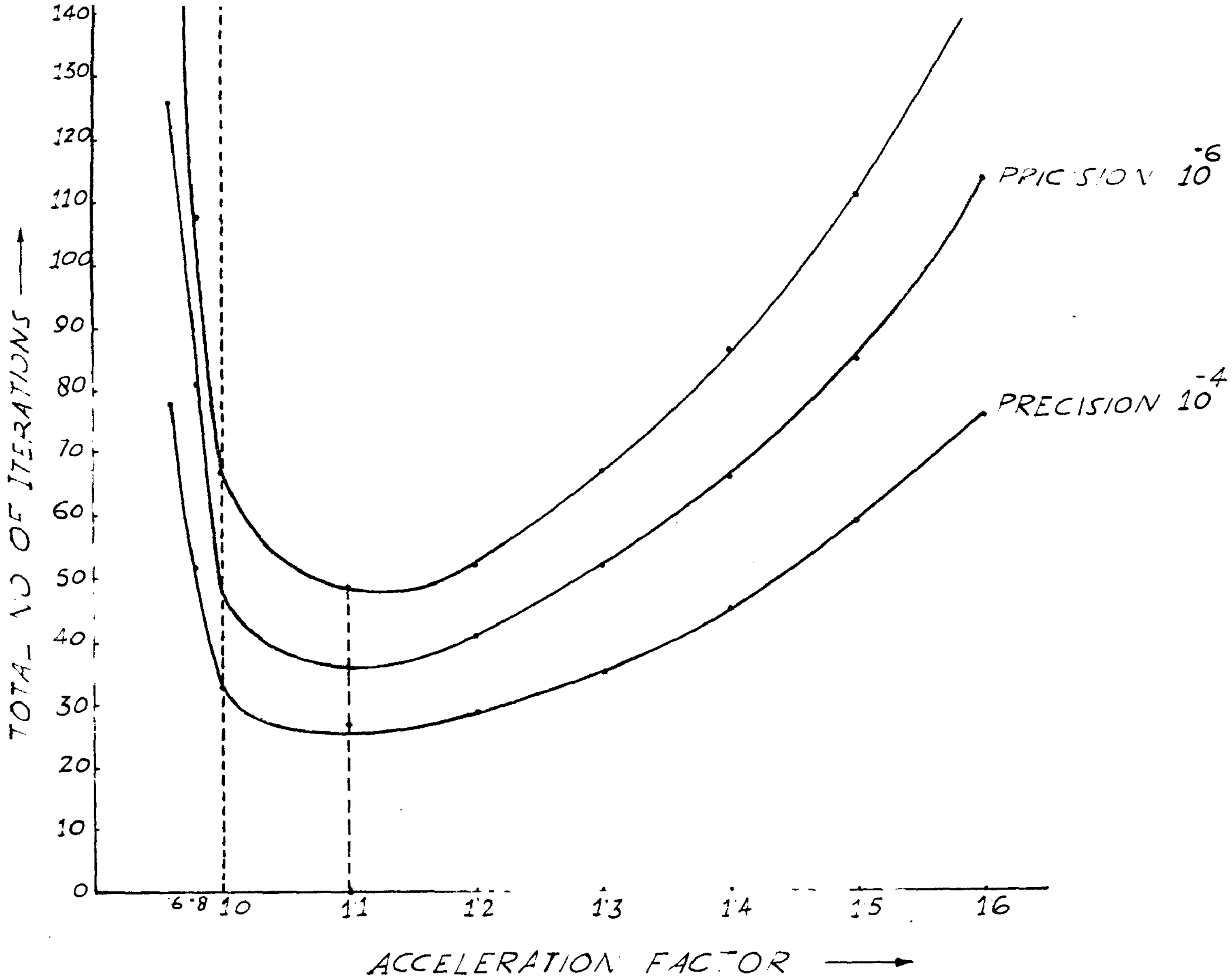

TOTAL NO OF ITERATIONS
0
10
20
30
40
50
60
70
80
90
100
110
120
130
140
·6
·8
1·0
1·1
1·2
1·3
1·4
1·5
1·6
ACCELERATION FACTOR
PRECISION 10^{-6}
PRECISION 10^{-4}

FIGURE 4-6

TABLE 4-4-4

5 BUS SYSTEM

Precision 10^{-4}

BUS NO	NO. OF ITERATIONS										
	Acceleration Factor	0.6	0.8	1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7
1		24	16	11	08	$\boxed{07}$	09	11	15	19	-
2		12	08	$\boxed{04}$	05	07	09	11	15	19	-
3		21	14	09	$\boxed{07}$	07	08	11	14	18	-
4		21	14	09	$\boxed{07}$	08	09	12	15	20	-
Total No. of Iternations	-	78	52	33	$\boxed{27}$	29	35	45	59	76	-

The above values are for a 5-bus system of Fig. 4-2

TABLE 4-4-3

5 BUS-SYSTEM

PRECISION 10^{-6}

BUS NO	NUMBER OF ITERATIONS										
	Acceleration Factor	0.6	0.8	1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7
1		39	25	16	11	$\boxed{09}$	13	16	21	28	-
2		19	12	$\boxed{05}$	07	10	13	16	21	28	-
3		35	22	14	$\boxed{10}$	11	13	17	22	29	-
4		33	22	13	$\boxed{08}$	11	13	17	21	29	-
Total No. of Iterations -		126	81	48	$\boxed{36}$	41	52	66	85	114	-

The above values are for 5-bus system of Figure 4.2

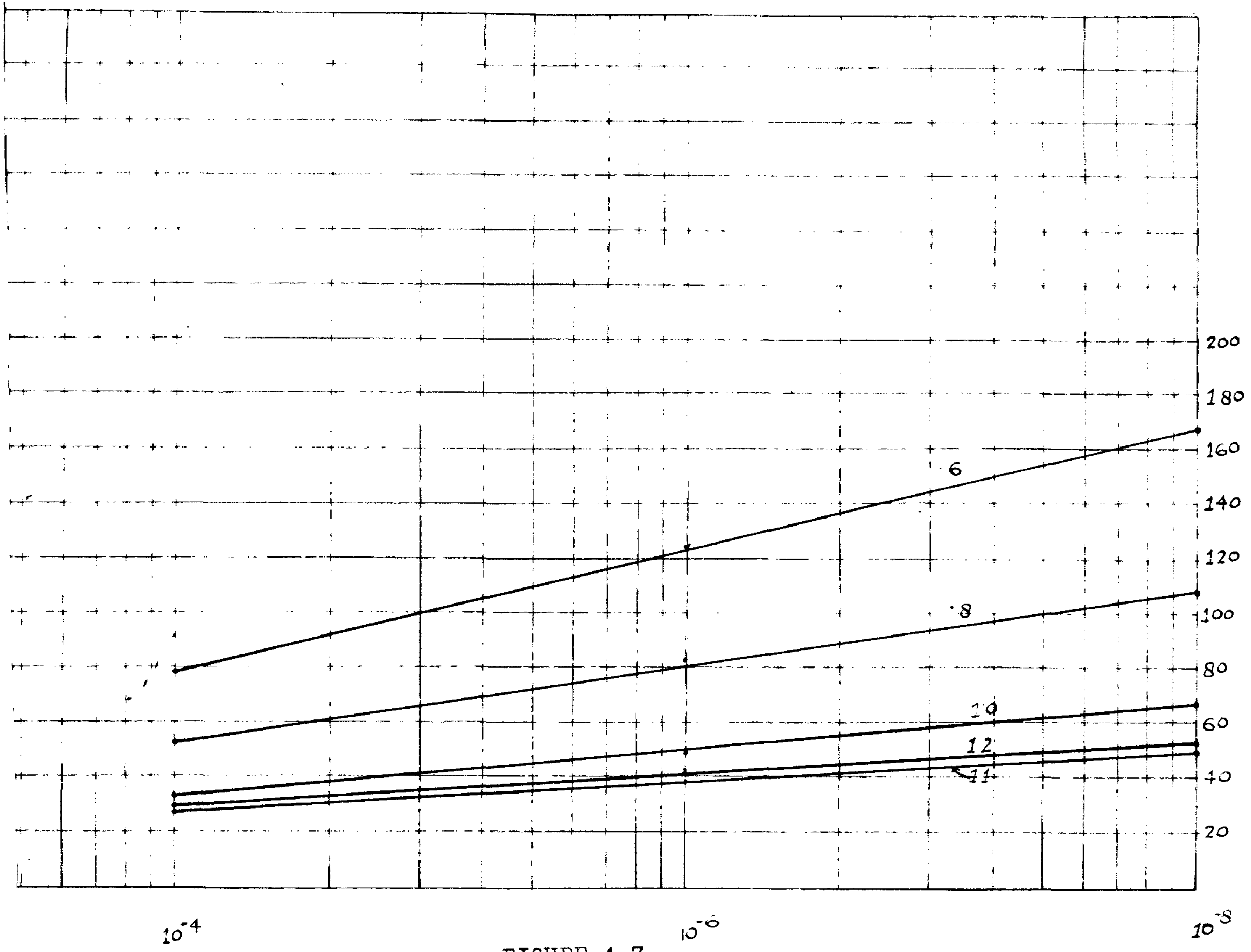

6
8
10
12
11
200
180
160
140
120
100
80
60
40
20
10^{-4}
10^{-6}
10^{-8}
PRECIS ON

FIGURE 4-7

TABLE 4.4-5

PRECISION 10^{-8} ; ACCELERATION FACTOR 1.3

5 BUS SYSTEM

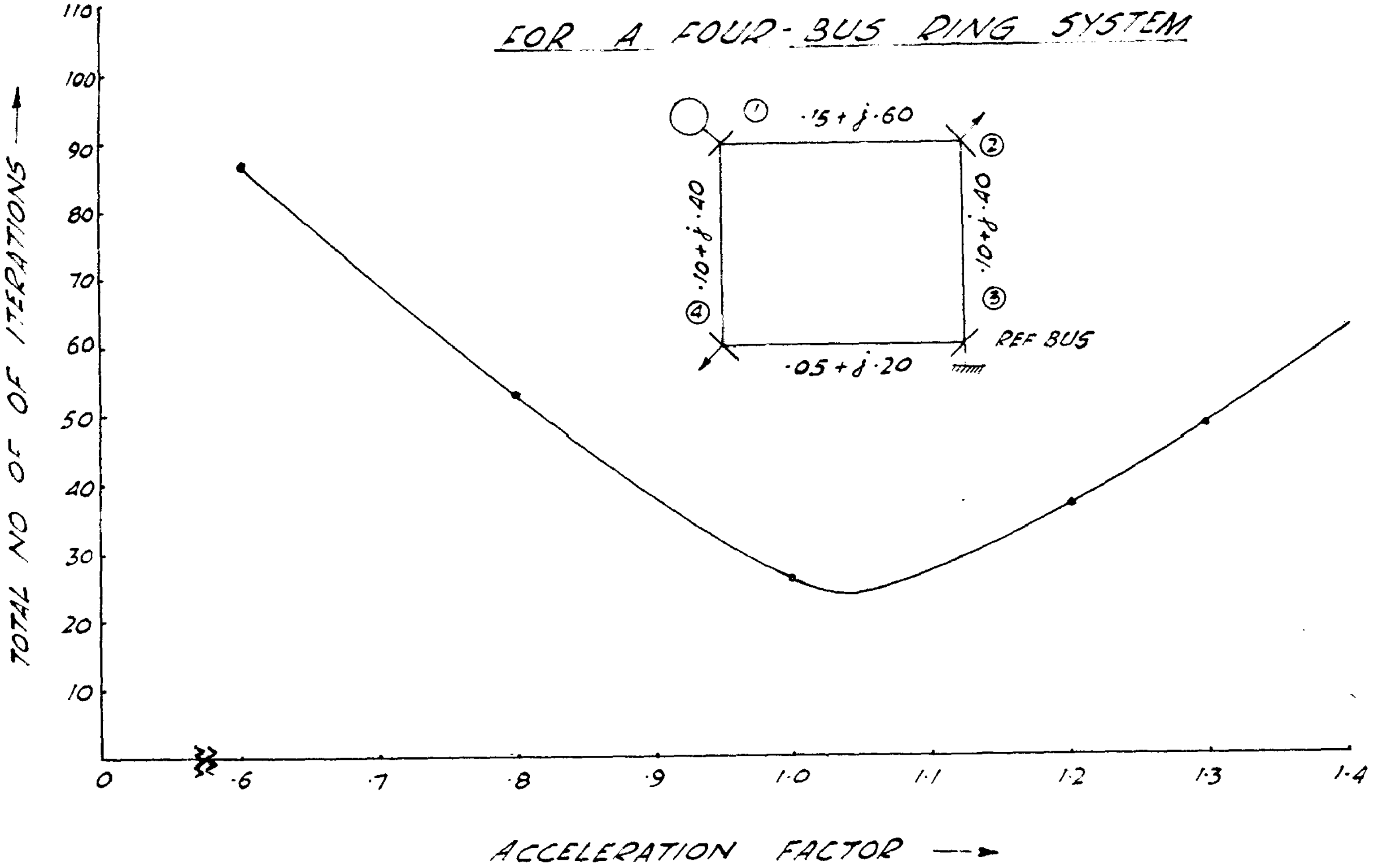
BUS NO	NO. OF ITERATIONS				
	Reference Bus -1	Reference Bus 2	Reference Bus 3	Reference Bus 4	Reference Bus 5
1	X	16	16	17	18
2	16	X	15	18	15
3	16	16	X	15	17
4	16	17	16	X	17
5	17	18	17	50*	X
Total No. of Iteration	65	67	64	100*	67

TABLE 4-4-6

A FOUR BUS RING SYSTEM: ACCURACY 10^{-8}

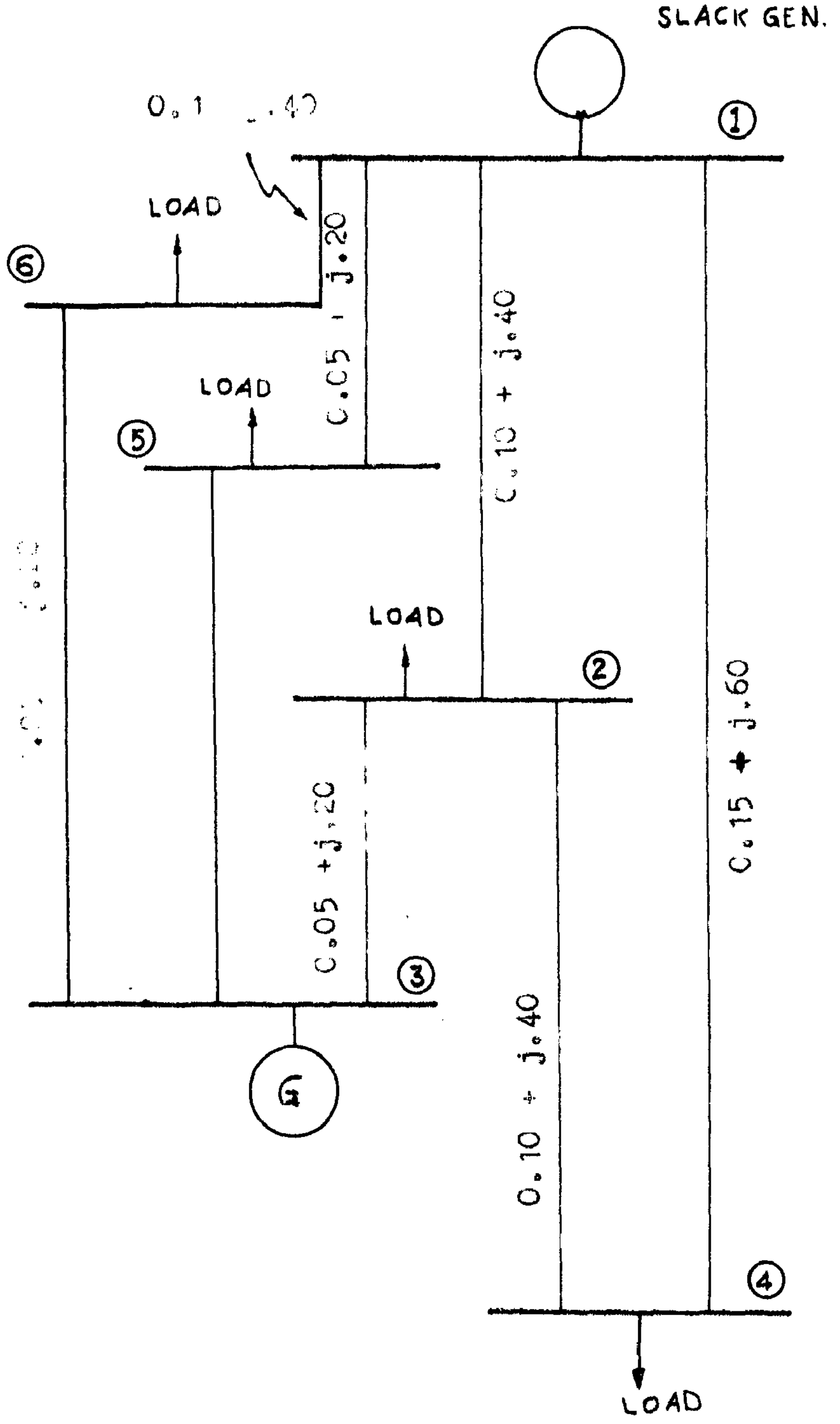
BUS 3 IS REFERENCE BUS

BUS NO	NUMBER OF ITERATIONS						
Acceleration factor	0.6	0.8	1.0	1.1	1.2	1.3	1.4
1	20	12	$\boxed{02}$	09	12	15	20
2	34	21	13	$\boxed{09}$	13	16	21
4	32	20	11	$\boxed{10}$	13	16	22
Total No. of Iterations:	86	53	$\boxed{26}$	28	38	47	63



FOR A FOUR-BUS RING SYSTEM

FIGURE 4.8



ONE LINE DIAGRAM

(Branch Impedance Shown)

FIGURE 4-9

TABLE 4-4-7

6 BUS SYSTEM CONTAINING 3 LOOPS; PRECISION 10^{-8}

6 BUS NO	NO. OF ITERATIONS							
	Acceleration factor -	.8	1.0	1.1	1.2	1.3	1.4	1.5
1		38	25	18	$\boxed{14}$	18	21	29
2		26	15	$\boxed{10}$	14	15	21	26
3		28	17	$\boxed{12}$	14	20	21	30
4		45	29	23	$\boxed{16}$	19	23	33
6		34	21	18	$\boxed{13}$	18	22	60
Total No.of Iterations		171	107	81	$\boxed{71}$	90	108	178

TABLE 4-4-8

ACCELERATION FACTOR OF 1.3

No. of Buses	Total No. of Iterations
2	0
4	47
5	67
6	90

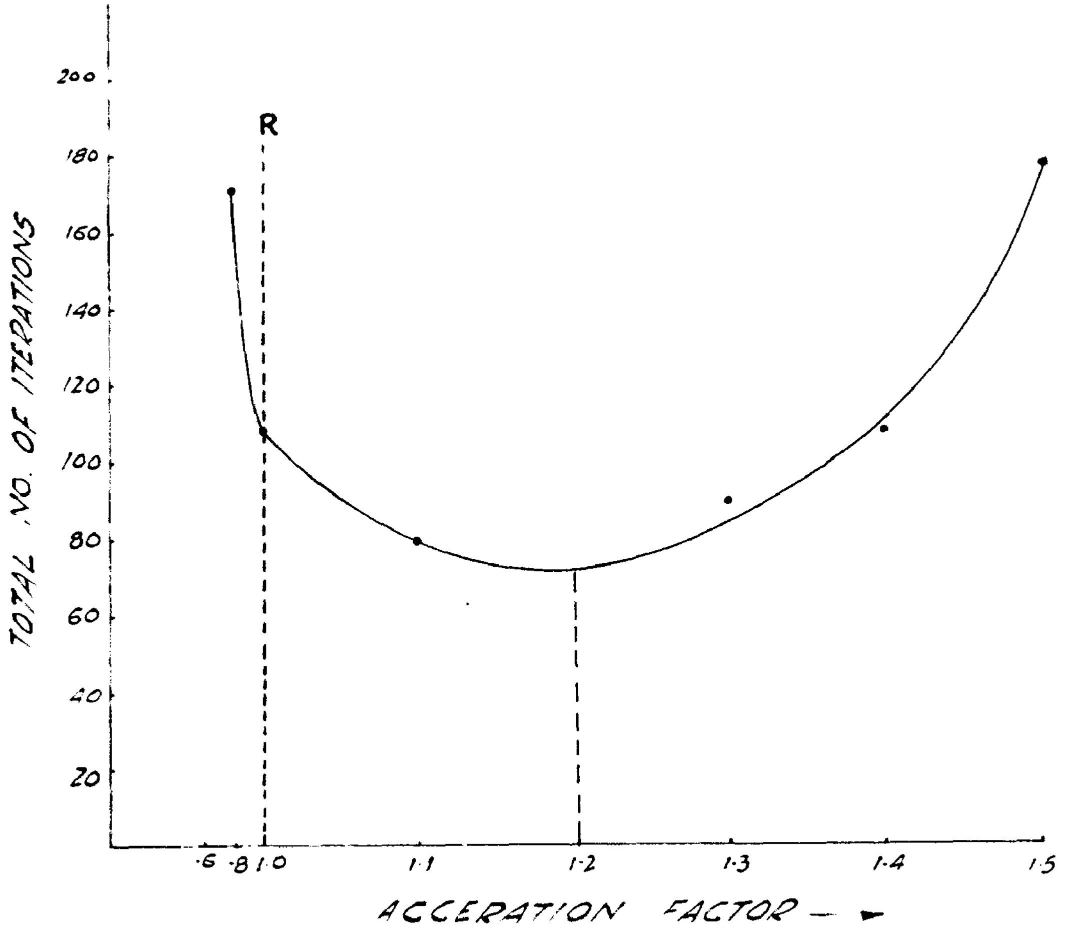


FIGURE 4-10

FOR A 6-BUS, 3 LOODS SYSTEM

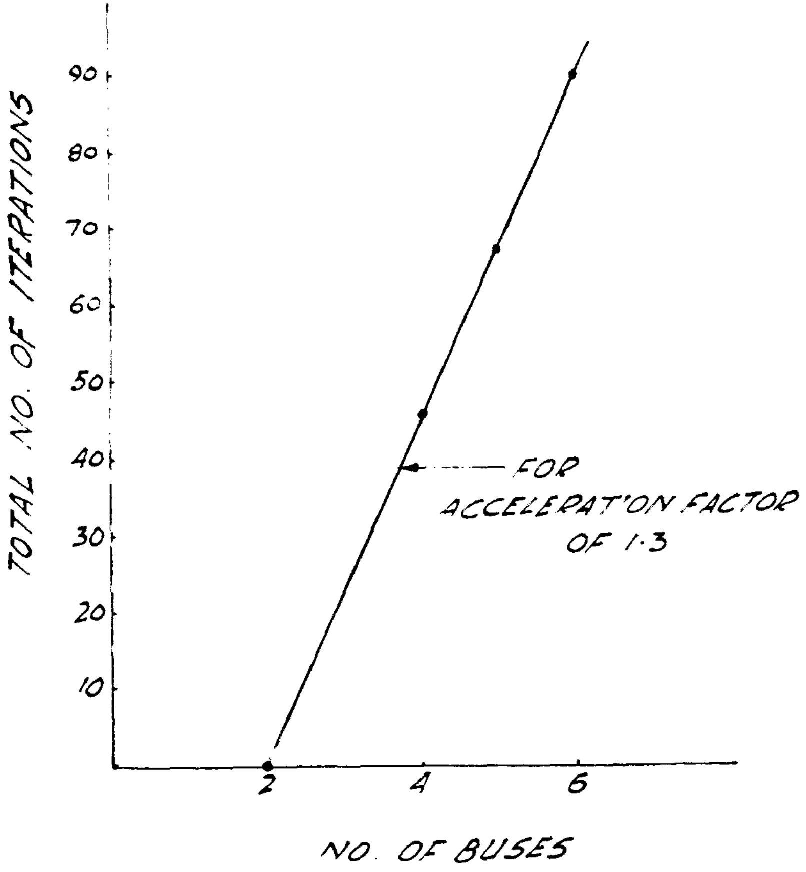


FIGURE 4-11

CHAPTER - 5

LOAD FLOW STUDIES

5.1. Introduction

The load or power flow studies consist of imposing specified power input and voltage magnitude or real and reactive power input conditions, at the terminals of a passive network under existing or contemplated conditions of normal operation. The solution provides complete input and voltage information at terminals and Power flow in each branch of the network.

Load studies are essential in planning the future development of the system because satisfactory operation of the system depends on knowing the effects of interconnections with other power system of new loads, new generating stations, and new transmission lines before they are actually installed.

5.2. Nature of the Problem

For any normal power system network the node equations can be written as:

$$I_k = \sum_{m=1}^{N} Y_{km} E_m \qquad \ldots (1)$$

where E_m represents the voltages at the nodes and I_k the current flowing in to the nodes. Y_{km} are the admittances.

The neutral or ground node is taken as a reference and all voltages are referred to it.

In a transmission network, the node current may either be coming from a generator or flowing to a load. If there were no generation or load such as, at a tie point represented by a node, the current would be zero. Since I_k , Y_{km}, E_m are complex quantities, they can be written as

$$I_k = I_k e^{j\alpha k} = a_k + jb_k$$

$$E_m = E_m e^{j\delta m} = e_m + jf_m$$

$$Y_{km} = Y_{km} e^{j\theta k_m} = G_{km} + jB_{km} \qquad (2)$$

The power equation then could be stated as (conjugating IX)

$$P_k + jQ_k = \sum_{m=1}^{N} Y_{km}^{*} E_m^{*} E_k \qquad (3)$$

$$\text{or } P_k + jQ_k = \sum_{m=1}^{N} Y_{km} E_m E_k \, e^{j(\delta k - \delta m - \theta_{km})} \qquad \text{----}(4)$$

where k is the number of the node

At each node there are four variables:

P_k, Q_k, E_k and δ_k, other quantities being known.

It is the varied data information available for different nodes, that complicates the otherwise straight forward solution. In any system there can be the following 3 types of nodes vis.

1. Slack bus; which has the magnitude E_k and the angle of its voltage δ_k specified, but the real and reactive powers, P_k and Q_k are not known. This bus makes up the difference between the scheduled

loads and generation that is caused by losses in transmission system.

2. Generator buses, The values of E_k and P_k are specified leaving δ_k and Q_k as unknowns

3. Load buses: The real and reactive powers, P_k, Q_k are specified and E_k, δ_k are unknowns.

5.3. <u>Possible solutions</u>

A formal, closed form of solution for even very simple this type of network may be almost difficult to obtain. There are analogue and digital methods and therefore the load flow studies could either be conducted on an a.c. network analyser or an digital computers. In an a.c. network analyser solution the first step is to set up components to represent the system, the line units adjusted to proper values of resistance, reactance and susceptance. The shunt capacitance of lines are taken in the account if desired. The auto-transformers are set for preliminary tap position and may be readjusted to give desired conditions. Voltage at slack bus is maintained at specified value, other generators being adjusted to specified voltage and real power conditions. Now the final solution requires successive adjustments of load and generators to converge on a simultaneous balance of all specified terminal conditions. Bus voltages and power flow are then read on the master instruments and recorded correspondingly on one-line diagram. Digital solution consists of using an iterative process by assuming initially some value of unknown voltages (magnitudes and their angles), the specified conditions are applied at each bus, and the correction is applied successively to each bus voltage till the voltages converge to an assigned precision. The line flow is then computed. This method is known as Nodal Iterative method (40,42).

A different approach to the power flow solution also exists. In that, the first step is the same i.e., to assume some voltages for each bus, where the voltages are not specified.

Next with the assumed set of voltages the power- input to each bus is computed and then the difference of the scheduled power and power calculated is found and necessary correction correspondingly is computed from the set of linearised difference equations. This process of finding the difference of power and then to find the voltage correction proceeds till the differences of power vanishes or the voltages converge. This method (47,49,50,51) supercedes the method described by Ward and Hale (40), originally.

5.4 Steps necessary for a digital Load flow studies

The following steps are necessary before a system can be taken for load flow studies, with single line diagram available.

1. Coding of the network, which consists of labelling the buses and branches.

The slack machine which supplies the difference between the specified real power in to the system at other buses and the total system output plus losses, is numbered as bus 1.

Other buses can be numbered arbitrarily and the reference or neutral bus may be left unnumbered. Branches are numbered sequencially.

2. Intermediate nodes may be eliminated at first.
3. Off nominal turns ratios are considered on the basis of analysis given in reference (40)
4. Self and mutual admittances are computed corresponding to each bus from the data available for branch impedances. This process can be computerised also and included with main load flow program as done in the programme written by the author in Appendix-F
5. With all the input data available in proper form any method such as Nodal Iterative method can be used for voltage solution or as described in reference (49).

5.5 Ward and Hale (40) Method

The method as suggested originally by Ward and Hale had the following stages.

1. The process is started with some assumed voltages ~~a~~for all buses except the slack generator in which case the value as given in input data is used throughout. The assumed values may be taken as 1 + j0.

2. These values assumed are used to calculate the impressed current at bus 2 using equation (1) with this value of current and assumed value of voltage at bus 2, the power input at bus 2 is calculated.
3. A correction is obtained in the value of voltage at bus 2 using the scheduled power and scheduled voltage magnitude or reactive power if the bus is a generator or load bus, other bus voltages are considered as constant while calculating the correction.
4. With corrected value of voltage at bus 2 and other bus voltages (assumed), the current input at bus 3 is found by the equation (1). Again with this current and voltage at bus 3 power is computed.
5. With this power input at 3 calculated, the step 3 is again repeated to find the correction in the value of voltage at bus 3.

This process is continued till all corrected bus voltages are known, which replaces the previous assumed set of voltages. Once this is done all the steps right from the step 1, are repeated again and again till the voltages correction becomes less than a preassigned precision.

There are different corrections to be applied at generator and load buses, so the identification is necessary to recognise a bus as a generator or load bus. This may automatically be done by computer as in the author's programme using an index to be attached with input data. After the voltages have converged, the power flow in each branch can be calculated and relevent information may be obtained as regards losses and mismatch.

5.6 Generalised Method as described in References (43, 46)

The Ward and Hale method has a drawback that the Corrections needed to be applied at generator and load buses are to be computed seperate~~d~~ly and for real and imaginary parts both. The process is slow.

However, a generalised method as described in reference (43) can be applied to find the converged voltages. The method can be described as follows

If I_1 be the input current and $P_1 - jQ_1$ be the input Power at bus 1.

Then I_1 can be written as

$$I_1 = \sum_{q=2}^{n} Y^{1q} \quad (E_1 - E_q) \quad \ldots\ldots \quad (5)$$

But $E_1^* \; I_1 = P_1 - jQ_1 \quad \ldots\ldots \quad (6)$

Therefore

$$\frac{P_1 - jQ_1}{E_1^*} = (Y^{12} + Y^{13} + Y^{14} + \ldots\ldots\ Y^{1n})\ E_1 \\ - Y^{12}E_2 - Y^{13}E_3 - Y^{14}E_4 \ldots\ldots\ Y^{1n}E_n\ ) .. (7)$$

Solving for E_1

$$E_1 = \frac{\dfrac{P_1 - jQ_1}{E_1^*} + Y^{12}E_2 + Y^{13}E_3 + \ldots\ Y^{1n}E_n}{Y^{12} + Y^{13} + Y^{1n}}$$

or in general

$$E_p = \frac{\dfrac{P_b - jQ_p}{E_p^*} + \sum_{q=1}^{n} Y^{pq}\ E_q}{\sum_{q=1}^{n} Y^{pq}} \qquad \text{for } q \neq p \quad \ldots (8)$$

Now this equation can be used for any bus with the following procedure.

1. At a load bus the real and reactive powers are specified so E_p can be found from equation (8) through an iterative process from some assumed values ~~fm~~ of voltages and then improving it in successive iterations.

2. At the generator bus where real power and constant voltage magnitude is specified the procedure for voltage iterations is as follows.

   (i) Since reactive power is not spcified so first Q_p is calculated from assumed voltages by

$$Q_p = - I_m \left[\left(\sum_{q=1}^{n} Y^{pq} \quad E_p + \sum_{q=1}^{n} Y^{pq} \quad E_q \right) \right] E_p^* \qquad \ldots(9)$$

where $q \neq P$

   (ii) after calculating Q_p and with specified P_p the voltage is calculated from (8) and reduced proportionately to correspond to the magnitude requirement.

Here, since in (8) E_p is a function of E_p or rather itself so before going to the next bus for iterations, resubstitution of new value of E_p in (8) as E_p^* is required to give further refined value of new E_p' which will be used for next iteration.

In all the above iterative processes, the number of iterations may be reduced considerably by projecting the value of E_p towards actual value by using some acceleration factor.

$$E'_{pi} = (E_{pi} - E_{pi-1})\alpha + E_{pi-1} \quad \ldots (10)$$

Where α is an acceleration factor. The above equation may be used for both real and imaginary parts of voltages. The choice of α depends again on the system under study. For a moderately sized system a choice of 1.6 and 1.7 for real and imaginary parts respectively is good. Once again, when the voltages have converged to a desired precision the power flow in each branch can be calculated from

$$P_{pq} + jQ_{pq} = E_p\,(E_p - E_q)^{*}\ (\ \ - Y^{pq})^{*} \quad \ldots.\ (11)$$

where P_{pq} and Q_{pq} are the real and reactive power flow from bus P towards q. The programme writeen by the author and as included in Appendix F, is based on this method.

Reference (46) gives modified and enlarged version of this method only.

5.7 Other Approaches to Load Flow Solutions

Reference (47) describes a little different approach to the load flow problem. In load flow problem it is necessary to solve the set of simultaneous non-linear equation as represented by equation (4)

By taking total differentials, the followig linear relationship is developed for small variations in the variables of equation (4)

$$\Delta P_k = \sum_{m=1}^{N} H_{km}\,\Delta\delta_m + \sum_{m=1}^{N} N_{km}\,\Delta E_m$$

$$\Delta Q_k = \sum_{m=1}^{N} J_{km} \Delta\delta m + \sum_{m=1}^{N} L_{km} \Delta E_m \qquad \ldots(12)$$

H_{km}, N_{km}, J_{km} and L_{km} are the coefficients given by

for $k \neq m$

$$H_{km} = \frac{\partial P_k}{\partial \delta_m} = a_m f_k - b_m e_k$$

$$N_{km} = \frac{\partial P_k}{\partial E_m} = \frac{a_m e_k + b_m f_k}{E_m}$$

$$J_{km} = \frac{\partial Q_k}{\partial \delta_m} = -(a_m e_k + b_m f_k)$$

$$L_{km} = \frac{\partial Q_k}{\partial E_m} = \frac{a_m f_k - b_m e_k}{E_m} \qquad \ldots (13)$$

For $k = m$

$$H_{kk} = \frac{\partial P_k}{\partial \delta_k} = - Q_k - B_{kk} E_k^2$$

$$N_{kk} = \frac{\partial P_k}{\partial E_k} = \frac{P_k}{E_k} + G_{kk} E_k$$

$$J_{kk} = \frac{\partial Q_k}{\partial \delta_k} = P_k - G_{kk} E_k^2$$

$$L_{kk} = \frac{\partial Q_k}{\partial E_k} = \frac{Q_k}{E_k} - B_{kk} E_k \qquad \ldots. (14)$$

Where am and b_m are the components given by

$$a_m + jb_m = (G_{km} + jB_{km})(e_m + jf_m)$$

and $\Delta\delta$'s are in radians.

For a 5-bus sustem of figure 5-1 the equations would be

$$\begin{vmatrix} \Delta P_2 \\ \Delta P_3 \\ \Delta P_4 \\ \Delta P_5 \\ \Delta Q_2 \\ \Delta Q_4 \\ \Delta Q_5 \end{vmatrix} = \begin{vmatrix} H_{22} & H_{23} & H_{24} & H_{25} & N_{22} & N_{24} & N_{25} \\ H_{32} & H_{33} & H_{34} & H_{35} & N_{32} & N_{34} & N_{35} \\ H_{42} & H_{43} & H_{44} & H_{45} & N_{42} & N_{44} & N_{45} \\ H_{52} & H_{53} & H_{54} & H_{55} & N_{52} & N_{54} & N_{55} \\ J_{22} & J_{23} & J_{24} & J_{25} & L_{22} & L24 & L_{25} \\ J_{42} & J_{43} & J_{44} & J_{45} & L_{42} & L_{44} & L_{45} \\ J_{52} & J_{53} & J_{54} & J_{55} & L_{52} & L_{54} & L_{55} \end{vmatrix} \begin{vmatrix} \Delta\delta_2 \\ \Delta\delta_3 \\ \Delta\delta_4 \\ \Delta\delta_5 \\ \Delta E_2 \\ \Delta E_4 \\ \Delta E_5 \end{vmatrix} \qquad -(15)$$

The coefficients listed above are not constant but are functions of the operating point, and thus of E_k and δ_k. In equation (15) bus 1 is not taken in to account because it is a slack bus, i.e. E_1 and δ_1 are fixed, hence ΔE_1 and $\Delta\delta_1$ are zero.

Similarly at bus 3, since it is a generator bus ΔE_3 is zero.

Now by comparing the actual real and reactive power at a bus with desired values the $\Delta P_{k's}$ and $\Delta Q_{k's}$ can be evaluated and equation (15) is solved for $\Delta\delta_m$'s and $\Delta E_{m's}$.

Equation (12) represents a set of linear equations that can be solved for desired corrections of the variables in equation (4). But because of the non-linearity, these correction would only be effective if they are small. Therefor, an approximate solution for equation (12) is found and using this to find a operating point for equation (4) and then from this new operating point, reevaluating the coefficents of equ tion (12) is done. This process is repeated till the desired solution is obtained. Generally, equation (12) is not solved more accurately because it might as well take long time to arrive at the solution, so more iterations are done on equation (4) rather than solving equation (12) correctly , so that desired solution is obtained in less time.

In the reference (47), exact solution of equation (15) has not been obtained and are solved by an iterative method but in reference (51), Van Ness suggests that equation (15) may be solved by the elimination method so that within round off error, the exact correction factors are found for all buses, before they are applied to the voltage magnitude and angle. Although the corrections may be exactly correct as far as equation (15) is concerned, they may not correct voltages and angles completely. However, they may bring results much closer.

In this method (51) with the assumed values of E andδ, P,s and Q's are calculated. Also the coefficient matrix of equation (15) is computed and stored. ΔP's and Qks ΔQ's are then found and equation (15) is solved by elimination method. The results are used to correct the voltages at buses and this process is repeated till the desired accuracy is obtained. The coefficient matrix of equation (15) is computed at each step as its elements are functions of the voltages.

5.8 Impedance Matrix Iterative Method

The driving and transfer impedances may also be used to define the network and thus an iterative process may consist of an initial assumption of currents, computation of voltages, determination of errors, in the terminal conditions and correction of currents to give new values to start the next iteration. All previous attempts in this direction were futile(48)This was perhaps due to lack of fast method of developing transfer impedance matrix. General Electric and Commonwealth Edison companies undertook this project and were successful in writing a programme based on above method as given in reference (52) of April 1963. In every system they studied time required for solution was much less than that required by the usual nodal branch admittance iterative programme.

Their programme essentially consists of three subroutines.

1. Matrix formation subroutine
2. Data sorting and modifying subroutine
3. Iterative subroutine.

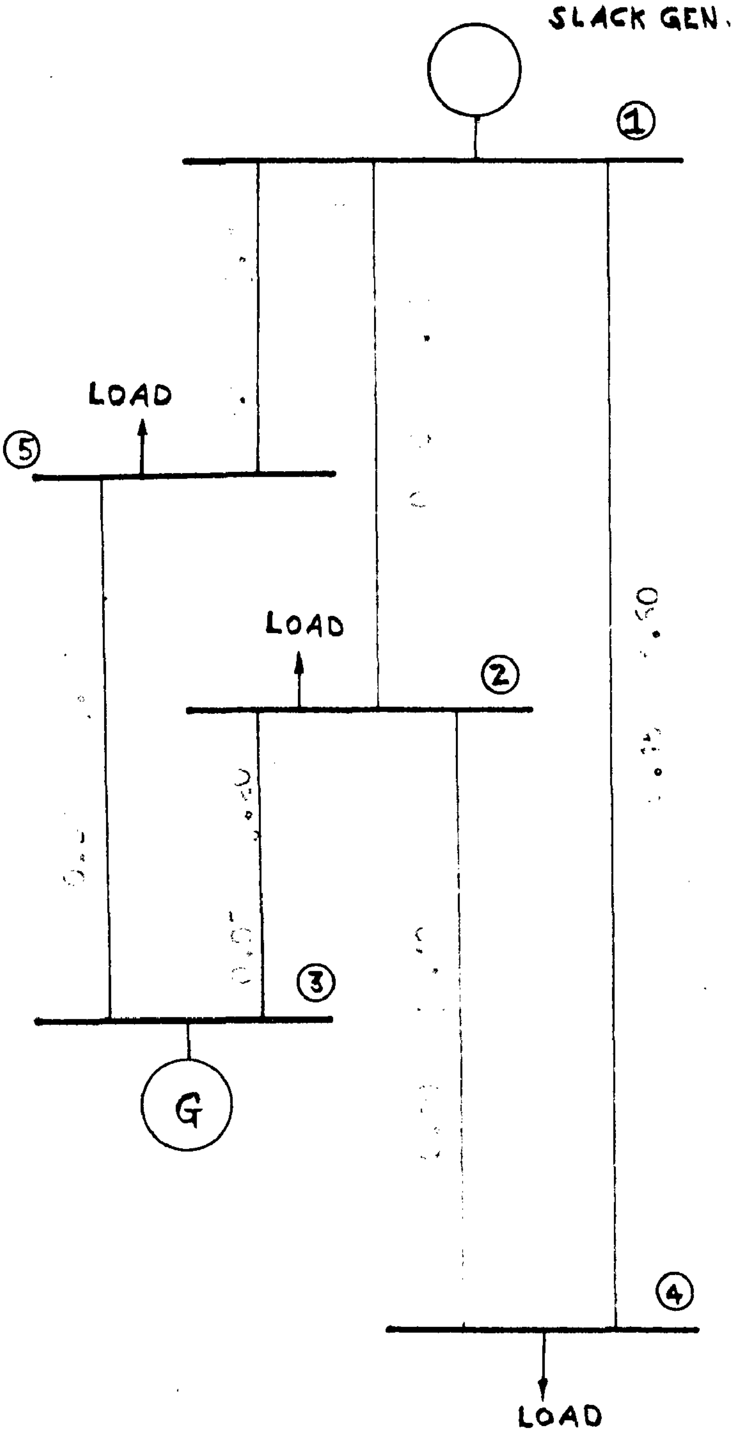
The first two subroutines prepare the Z-matrix so that it is quite adaptable to an iterative process. The details of this method are available in reference (52)

5.9 Illustrative Problems

The problems as shown in figure 5-1 was selected for the load flow problem. The input and output data is given below in the form of tables in compact form viz Table 5.9.1, 5.9.2, 5.9-3 and 5.9-4.

Table 5.9-1 INPUT DATA

BUS	P p.u	Q p.u	E p.u	Remarks
1	...	...	$1.02 \angle 0°$	Single bus
2.	0.6	0.3	$1.00\angle\underline{0}°$ assumed	Load bus
3	1.0	...	$1.04\angle\underline{0°}$	Generator bus, voltage magnitude constant
4	0.4	0.1	$1.00\angle\underline{0°}$ assumed	Load bus
5.	0.6	0.2	$1.00\angle\underline{0°}$ assumed	Load bus



ONE LINE DIAGRAM

BRANCH IMPEDANCES SHOWN

FIGURE 5-1

TABLE 5-9-2 **OUTPUT DATA (Self and Mutual Admittances)**

Y_{11} = 2.15686 - j8.6 2745

Y_{12} = -.58823 + j 2.35294

Y_{14} = -.39215 + j 1.56862

Y_{15} = -1.17647 + j 4.70588

Y_{22} = 2.35294 - j9.41176

Y_{23} = -1.17647 + j 4.70588

Y_{24} = -5.88235 + j 2.35294

Y_{33} = 2.35294 - j 9.41176

Y_{35} = -1.17647 + j 4.70588

Y_{44} = 0.98039 - j3.92156

Y_{55} = 2.35294 - j 9.41176

Here also $Y_{ij} = Y_{ji}$

TABLE 5-9-3: **OUTPUT DATA**

Bus	Voltage		Power		Remarks
	Mag	Angle	P	Q	
1	1.02	0.00	0.6515	0.3293	Slack bus
2	0.9547	-3.94	0.6000	0.3000	Load Bus
3	1.04	2.00	1.0000	0.4769	Generator Bus
4	0.9234	-8.01	0.4000	0.1000	Load bus
5	0.9931	-2.07	0.6000	0.2000	Load bus

TABLE 5-9-4 : POWER FLOWS

BUSES From	To	P p.u.	Q p.u.
1	2	0.1980	0.1227
1	4	0.2481	0.1175
1	5	0.2054	0.0891
2	1	-0.1928	-0.1018
2	3	-0.5732	-0.2370
2	4	0.1661	0.0388
3	2	0.5943	0.3214
3	5	0.4057	0.1555
4	1	-0.2373	-0.0740
4	2	-0.1629	-0.0260
5	1	-0.2030	-0.0795
5	3	-0.3970	-0.1206

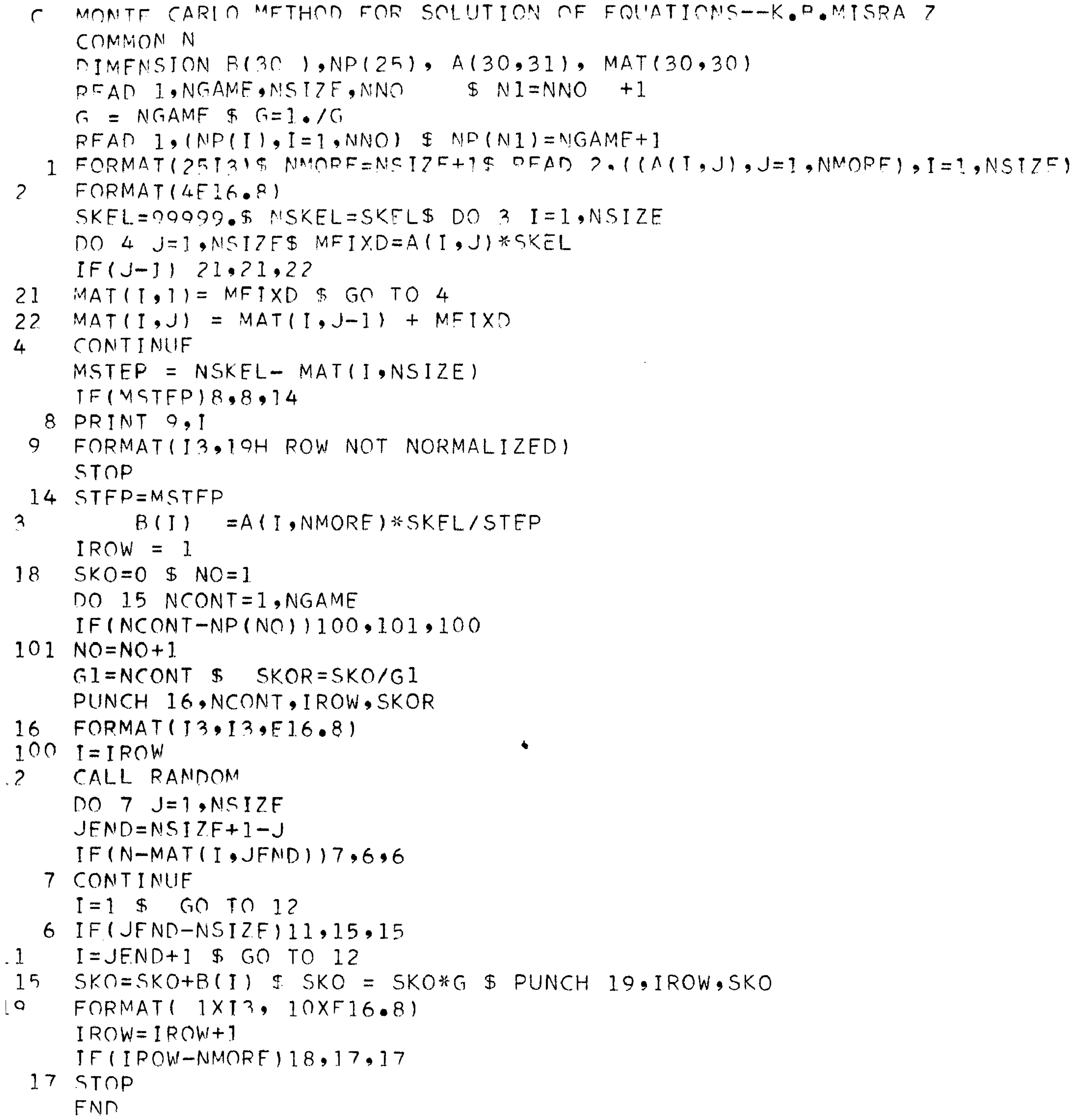
5-10 Further Proposed Work

The author wishes to modify the programme given in Appendix F and to improve upon the same. The programme based on the method given in reference (47) would also be tried and studied as regards speed of convergence and other factors.

The author feels that, since the exact solution of equation (15) is not desired very much for applying corrections to voltages and angles, the Monte Carlo method of solution of linear equations may be of help in handling the problem and perhaps time of computation may be less with this approach.

APENDIX

APPENDIX - A



```
C    MONTE CARLO METHOD FOR SOLUTION OF EQUATIONS--K.P.MISRA 7
     COMMON N
     DIMENSION B(30 ),NP(25), A(30,31), MAT(30,30)
     READ 1,NGAME,NSIZE,NNO      $ N1=NNO   +1
     G = NGAME $ G=1./G
     READ 1,(NP(I),I=1,NNO) $ NP(N1)=NGAME+1
   1 FORMAT(25I3)$ NMORE=NSIZE+1$ READ 2,((A(I,J),J=1,NMORE),I=1,NSIZE)
2    FORMAT(4E16.8)
     SKEL=99999.$ NSKEL=SKEL$ DO 3 I=1,NSIZE
     DO 4 J=1,NSIZE$ MFIXD=A(I,J)*SKEL
     IF(J-1) 21,21,22
21   MAT(I,1)= MFIXD $ GO TO 4
22   MAT(I,J) = MAT(I,J-1) + MFIXD
4    CONTINUE
     MSTEP = NSKEL- MAT(I,NSIZE)
     IF(MSTEP)8,8,14
   8 PRINT 9,I
  9  FORMAT(I3,19H ROW NOT NORMALIZED)
     STOP
 14  STEP=MSTEP
3        B(I)   =A(I,NMORE)*SKEL/STEP
     IROW = 1
18   SKO=0 $ NO=1
     DO 15 NCONT=1,NGAME
     IF(NCONT-NP(NO))100,101,100
101  NO=NO+1
     G1=NCONT $   SKOR=SKO/G1
     PUNCH 16,NCONT,IROW,SKOR
16   FORMAT(I3,I3,E16.8)
100  I=IROW
12   CALL RANDOM
     DO 7 J=1,NSIZE
     JEND=NSIZE+1-J
     IF(N-MAT(I,JEND))7,6,6
   7 CONTINUE
     I=1 $   GO TO 12
   6 IF(JEND-NSIZE)11,15,15
11   I=JEND+1 $ GO TO 12
15   SKO=SKO+B(I) $ SKO = SKO*G $ PUNCH 19,IROW,SKO
19   FORMAT( 1XI3, 10XF16.8)
     IROW=IROW+1
     IF(IROW-NMORE)18,17,17
 17  STOP
     END
```

APPENDIX - B

```
C   C  ITERATIVE METHOD K.B.MISRA    Z
        DIMENSIONA(20,20),B(20),X(20)
        DO50I=1,20
        B(I)=0.0$ X(I)=0.0
        DO50J=1,20
   50   A(I,J)=0.0
        READ20,N,TEST
   20   FORMAT(I2,F10.6)
   31  READ21,I,J,A(I,J),I1,J1,A(I1,J1),NEXT
   21   FORMAT(2(2I2,F10.6),I1)
        GOTO(31,32),NEXT
   32   READ22,I,B(I),NEXT
   22   FORMAT(I2,F10.6,I1)
         GOTO(32,33),NEXT
   33   READ22,I,X(I),NEXT
        GOTO(33,34),NEXT
   34   ERROR=0.0
        DO35I=1,N
        SUM=0.0
        DO40J=1,N
   40    SUM=SUM+A(I,J)*X(J)
   39    TEMPX=(B(I)-SUM+A(I,I)*X(I))/A(I,I)
        ERROR=ERROR+ABSF(X(I)-TEMPX)
   35    X(I)=TEMPX
       PUNCH23,ERROR
   23   FORMAT(6HERROR=,E20.8)
        IF(ERROR-TEST)41,34,34
   41    NEXT1=1
        NEXT2=2$ NM1=N-1
        PUNCH25
   25  FORMAT(18HTHE UNKNOWNS ARE- )
         DO42I=1,NM1
   42  PUNCH24,I,X(I),NEXT1
       PUNCH24,N,X(N),NEXT2
   24  FORMAT(1HE,I2,F12.8,I3)
        STOP
        END
```

APPENDIX - C

```
C   C   MONTE CARLO METHOD FOR LAPLACE EQUATION   K.B.M.   Z
        DIMENSION A(45,45),ANU(45,45)
        COMMON K
        READ 100, NG
100     FORMAT (I5)
        READ200,((A(I,J),J=1,8),I=1,8)
200     FORMAT  (8F9.2)
        DO 400 I=2,7 $ J=1
11      IF(A(I,J))13,12,13
12      J=J+1
        GO TO 11
13      J=J+1
        IF(J-8)14,400,400
14      IF(A(I,J)) 13,15,13
15      B=0. $ IG=1
16      IK=I $ JK=J
17      CALL RANDOM
        M=K/10000
        IF(M) 17,17,18
18      IF(M-5)19,17,17
19      GO TO (20,21,22,23),M
20      JK=JK+1
        GO TO 24
21      IK=IK-1
        GO TO 24
22      JK=JK-1
        GO TO 24
23      IK=IK+1
24      IF (A(IK,JK)) 25,17,25
25      B=B+A(IK,JK)
        IG=IG+1
        IF(IG-NG) 16,16,26
26      ANG=NG
        ANU(I,J) =B/ANG
        PUNCH300,I,J,ANU(I,J)
300     FORMAT(4HANU(,I2,1H,,I2,2H)=,E14.7)
        GOTO13
400     CONTINUE
        STOP
        END
```

```
SUBROUTINE RANDOM


N9415544565400000000080500000000000**                             Z000000000000001
J310243000K3320009000000220009900000 89K610243000993200095000000Z0   J0252J0312000002
26599990009942000000000Z                                          0   J0312J0336000003
J234056789Z                                                       1   J0234J0244000004
00000000000                                                       1J00000000000005


DATA FOR THE LAPLACE EQUATION


0.        0.        0.        0.        1.        1.        1.         1.
0.        0.        1.        1.        1.        0.        0.     .75
0.        0.        1.        0.        0.        0.        0.          .5
0.        0.        1.        0.        0.        0.        0.       .25
1.        1.        1.        0.        0.        0.     .00001   .00001
.67       0.        0.        0.        0.        0.     .00001        0.
.33       0.        0.        0.        0.        0.    .00001         0.
.00001   .00001   .00001   .00001   .00001   .00001   .00001   .00001
```

APPENDIX - D

APPENDIX - E

```
C   C   IMPEDANCES - K.B.MISRA    Z
        DIMENSIONYR(20,20),YI(20,20),ER(20),EI(20),ER1(20),EI1(20)
        COMMON N,YR,YI,ER,EI,ER1,EI1
        READ10,N,ITMAX,NRB,INDEX,CF,CONLM
        CALL INITAL
        CALL INPUTS
        DO1I=1,N
        ER(I)=0.
      1 EI(I)=0.
        DO9K=1,N
        IF(K-NRB) 13,11,13
     11 PUNCH 12,NRB
     12 FORMAT(6HBUS NO,I3,17H IS REFERENCE BUS)
        GO TO 9
     13 ER(K)=1.
        EI(K)=0.
        ITER=0
      2 CALL STORES
        ITER=ITER+1
        CALL VLTAGE(K,NRB,CF)
        CALL TESTER(CONLM,NS)
        GOTO(4,3),NS
      3 IF(ITER-ITMAX)2,4,4
      4 GOTO(8,7,6,5),INDEX
      5 CALL OUTPTA(K,ITER)
        GOTO9
      6 CALL CURENT(CINPR,CINPI,K)
        CALL OUTPTA(K,ITER)
        CALL OUTPTB(CINPR,CINPI,K)
        GOTO9
      7 CALL CURENT(CINPR,CINPI,K)
        CALL OUTPTA(K,ITER)
         CALL OUTPTB(CINPR,CINPI,K)
        CALL IMPDNC(CINPR,CINPI,K)
        GOTO9
      8 CALL CURENT(CINPR,CINPI,K)
        CALL IMPDNC(CINPR,CINPI,K)
      9 CONTINUE
 10     FORMAT(4I2,2F20.8)
        END
        SUBROUTINE INITAL
        DIMENSIONYR(20,20),YI(20,20),ER(20),EI(20),ER1(20),EI1(20)
        COMMON N,YR,YI,ER,EI,ER1,EI1
        DO1I=1,N
        DO1J=1,N
        YR(I,J)=0.
      1 YI(I,J)=0.
        RETURN
        END
```

```
  SUBROUTINE INPUTS
  DIMENSIONYR(20,20),YI(20,20),ER(20),EI(20),ER1(20),EI1(20)
  COMMON N,YR,YI,ER,EI,ER1,EI1
  DIMENSION IK(2), JK(2), YRK(2), YIK(2)
  READ 7, NCARD
  NC=0
1 READ 8,(IK(K),JK(K),YRK(K),YIK(K),K=1,2)
  NC=NC+1
2 DO5K=1,2
  I=IK(K)$ J=JK(K)
  IF(I)3,5,3
3 IF(J)4,5,4
4 YR(I,J)=YRK(K)
  YI(I,J)=YIK(K)
  YR (J,I)=YR(I,J)
  YI(J,I)=YI(I,J)
5 CONTINUE
  IF(NC-NCARD) 1,6,6
6 RETURN
7 FORMAT(I3)
8 FORMAT(2(2I2,2F18.8))
  END
  SUBROUTINE STORES
  DIMENSIONYR(20,20),YI(20,20),ER(20),EI(20),ER1(20),EI1(20)
  COMMON N,YR,YI,ER,EI,ER1,EI1
  DO1I=1,N
  ER1(I)=ER(I)
1 EI1(I)=EI(I)
  RETURN
  END
  SUBROUTINE VLTAGE(NEB,NRB, CF)
  DIMENSIONYR(20,20),YI(20,20),ER(20),EI(20),ER1(20),EI1(20)
  COMMON N,YR,YI,ER,EI,ER1,EI1
  DO6I=1,N
  IF(I-NEB)1,6,1
1 IF(I-NRB) 2,6,2
2 SUMNR=0.
  SUMNI=0.
  SUMDR=0.
  SUMDI=0.
  DO5J=1,N
  YRIJ=YR(I,J)
  YIIJ=YI(I,J)
  ERJ=ER(J)
  EIJ=EI(J)
  IF(YRIJ) 4,3,4
3 IF(YIIJ)4,5,4
4 CALL AMULTB(YRIJ,YIIJ,ERJ.EIJ,XR,XI)
  CALL APLUSB(SUMNR,SUMNI,XR,XI,SUMNR,SUMNI)
   CALL APLUSB (SUMDR,SUMDI,YRIJ,YIIJ,SUMDR,SUMDI)
5 CONTINUE
```

```
      CALL AUPONB(SUMNR,SUMNI,SUMDR,SUMDI,ER2,EI2)
      ER(I)=(ER2-ER(I))* CF+ER(I)
      EI(I)=(EI2-EI(I))* CF+EI(I)
    6 CONTINUE
      RETURN
      END
      SUBROUTINE TESTER (CONLM,NS)
      DIMENSIONYR(20,20),YI(20,20),ER(20),EI(20),ER1(20),EI1(20)
      COMMON N,YR,YI,ER,EI,ER1,EI1
      DO2I=1,N
      CR=ER(I)-ER1(I)
      CI=EI(I)-EI1(I)
      IF(ABSF(CR)-CONLM) 1,1,3
    1 IF(ABSF(CI)-CONLM) 2,2,3
    2 CONTINUE
      NS=1
      RETURN
    3 NS=2
      RETURN
      END
      SUBROUTINE OUTPTA(K,ITER)
      DIMENSIONYR(20,20),YI(20,20),ER(20),EI(20),ER1(20),EI1(20)
      COMMON N,YR,YI,ER,EI,ER1,EI1
      PUNCH 1, K
    1 FORMAT(//34HCONVERGED VOLTAGES CORR. TO BUS NO,I3,10H ENERGIZE
      NX=ITER-1
      PUNCH 2, ITER,NX
    2 FORMAT(/15X,5HAFTER,I3,11H ITERATIONS,17X,5HAFTER,I3,11H ITERA
     1S)
      PUNCH 3
    3 FORMAT(/6HBUS NO,8X,4HREAL,14X,4HIMAG,14X,4HREAL,14X,4HIMAG/)
      DO 5 I=1,N
    5 PUNCH 4,I,ER(I),EI(I),ER1(I),EI1(I)
   4  FORMAT(I4,F17.8,3F18.8)
      RETURN
      END
      SUBROUTINE CURENT(CINPR,CINPI,K)
      DIMENSIONYR(20,20),YI(20,20),ER(20),EI(20),ER1(20),EI1(20)
      COMMON N,YR,YI,ER,EI,ER1,EI1
      CINPR=0.
      CINPI=0.
      DO3I=1,N
      YRIJ=YR(K,I)
      YIIJ=YI(K,I)
      IF(YRIJ)2,1,2
    1 IF(YIIJ)2,3,2
    2 ERM=-ER(I)
      EIM=-EI(I)
      CALL APLUSB(1.,0.,ERM,EIM,XR,XI)
      CALL AMULTB(YRIJ,YIIJ,XR,XI,AR,AI)
      CALL APLUSB(CINPR,CINPI,AR,AI,CINPR,CINPI)
    3 CONTINUE
      RETURN
      END
      SUBROUTINE OUTPTB(CINPR,CINPI,K)
      DIMENSIONYR(20,20),YI(20,20),FR(20),FI(20),FR1(20),FI1(20)
      COMMON N,YR,YI,FR,FI,FR1,FI1
      PUNCH 1, K
    1 FORMAT(//29HINPUT CURRENT CORR. TO BUS NO,I3)
      PUNCH 2
    2 FORMAT(/14X,4HREAL,14X,4HIMAG/)
      PUNCH 3, CINPR, CINPI
    3 FORMAT(F21.8,F18.8)
      RETURN
      END
      SUBROUTINE IMPDNC (CINPR,CINPI,K)
      DIMENSIONYR(20,20),YI(20,20),FR(20),FI(20),FR1(20),FI1(20)
      COMMON N,YR,YI,FR,FI,FR1,FI1
      PUNCH 1,K
    1 FORMAT(//47HSELF AND MUTUAL IMP. AND ADM. CORRES. TO BUS NO,I3
      PUNCH 5
    5 FORMAT(//3X,1HI,3X,1HJ,14X,6HZ(I,J),31X,6HY(I,J))
      PUNCH 2
    2 FORMAT(/2(14X,4HREAL,14X,4HIMAG)/)
      DO 3 I=1,N
      CALL AUPONB(FR(I),FI(I),CINPR,CINPI,ZR,ZI)
      IF(ZR)10,11,10
   11 IF(ZI)10,12,10
   12 PUNCH13,K,I,ZR,ZI
   13 FORMAT(2I4,F13.8,F18.8,18X,8HINFINITE)
      GOTO3
   10 CALL AUPONB(1.,0.,ZR,ZI,TR,TI)
      PUNCH4,K,I,ZR,ZI,TR,TI
    3  CONTINUE
    4 FORMAT(2I4,F13.8,3F18.8)
      RETURN
      END
      SUBROUTINE AMULTB(AR,AI,BR,BI,ABR,ABI)
       ABR=AR*BR-AI*BI
      ABI=AR*BI+AI*BR
      RETURN
      END
      SUBROUTINE AUPONB(AR,AI,BR,BI,ABYBR,ABYBI)
      C=BR*BR+BI*BI
      ABYBR=(AR*BR+AI*BI)/C
      ABYBI=(AI*BR-AR*BI)/C
      RETURN
      END
      SUBROUTINE APLUSB(AR,AI,BR,BI,APLBR,APLBI)
      APLBR=AR+BR
      APLBI=AI+BI
      RETURN
      END
```

```
:  C  ELECTRICAL LOAD FLOW SOLUTION K.B.MISRA Z
      DIMENSION YR(20,20),YI(20,20),ER(20),EI(20),ER1(20),EI1(20)
      DIMENSION IN(20),IK(3),JK(3),RK(3),XK(3),P(20),Q(20),VMAG(20)
      DIMENSION BMTR(20),BMTI(20)
      COMMON ER,EI,ERS,EIS,IN,VMAG,SUMR,SUMI,J,N,I,YR,YI,QI,PI
      COMMON  ERI,EII,IK,JK,RK,XK,ER1,EI1,P,Q
      READ 20,N,ERS,EIS
      READ 21,CFR,CFI
      READ 22, CONLM
      PUNCH 101, CFR,CFI,CONLM
.01   FORMAT(3F20.8//)
      CALL INPTRX
      DO 1 I=2,N
.     IN(I)=0
      CALL INPTGL
      DO 3 I=2,N
      IF(IN(I))3,2,3
:     PRINT 23, I
      GO TO 30
)     CONTINUE
      CALL INITAL
      ITER=0
.0    DO4 I=2,N
      ER1(I)=ER(I)
,     EI1(I)=EI(I)
      DO9 I=2,N
      PI=P(I)
      ERI=ER(I) $ EII=EI(I)
      INI=IN(I)
      CALL SUMMAT
      GO TO (5,6),INI
)     QI=Q(I)
      CALL VOLTAG
      ER2=ERI $ EI2=EII
      CALL VOLTAG
      ER(I)=(ERI-ER2)*CFR+ER2
      EI(I)=(EII-EI2)*CFI+EI2
      GO TO9
)     CALL GENERQ(QRI)
      CALL VOLTAG
      IF(SENSE SWITCH 2)7,8
'     ERI=(ERI-ER(I))*CFR+ER(I) $ EII=(EII-EI(I))*CFI+EI(I)
      VMAGT=SQRTF(ERI*ERI+EII*EII) $ VM=VMAG(I)/VMAGT
      ER(I)=ERI*VM $ EI(I)=EII*VM
      GO TO 9
)      VMAGT=SQRTF(ERI*ERI+EII*EII) $ VM=VMAG(I)/VMAGT
      ERI=ERI*VM $ EII=EII*VM
      ER(I)=(ERI-ER(I))*CFR+ER(I) $ EI(I)=(EII-EI(I))*CFI+EI(I)
)     CONTINUE
      DO 12 I=2,N
      DER=ER(I)-ER1(I) $ DEI=EI(I)-EI1(I)
      IF(ABSF(DER)-CONLM)11,11,50
.1    IF(ABSF(DEI)-CONLM)12,12,50
.2    CONTINUE
```

```
      GO TO 31
50    IF(SENSE SWITCH 3)51,52
51    PRINT 100,ITER,DER,DEI
100   FORMAT(I3,2F20.8)
52    ITER=ITER+1 $ GO TO 10
31    I=1
      CALL SUMMAT
      ERI=ER(I) $ EII=EI(I)
      CALL GENERQ(QRI)
      P(1)=QRI $ Q(1)=QI
      DO 14 I=2,N
      INI=IN(I) $ ERI=ER(I) $ EII=EI(I)
      GO TO(14,13),INI
13    CALL SUMMAT $ CALL GENERQ(QRI) $ Q(I)=QI
14    CONTINUE
      PUNCH 24,ITER
      DO 15 I=1,N
      ERI=ER(I) $ EII=EI(I)
      VMAGT=SQRTF(ERI*ERI+EII*EII)
      ANGLE=(ATANF(EII/ERI))*180./3.14159
      PI=P(I) $ QI=Q(I)
      IF(I-1)62,66,62
62    INI=IN(I)
      GO TO (64,65),INI
64    PI=-P(I) $ QI=-Q(I)
      PUNCH 201,I,VMAG,ANGLE,PI,QI
      GO TO 15
65    VMAGT=VMAG(I)
66    PUNCH 202,I,VMAGT, ANGLE,PI,QI
15    CONTINUE
      PUNCH 26
      CALL BRADMT
      DO 53 I=1,N
      BMR=0. $ BMI=0.
      DO 19 J=1,N
      IF(I-J)16,19,16
16    YRIJ=YR(I,J) $ YIIJ=YI(I,J)
      IF(YRIJ)18,17,18
17    IF(YIIJ)18,19,18
18    DR=ER(I)-ER(J) $ DI=EI(I)-EI(J)
      CALL AMULTB(DR,DI,YRIJ,YIIJ,DR,DI)
      DI=-DI
      CALL AMULTB(ER(I),EI(I),DR,DI,PI,QI)
      PUNCH 27,I,J,PI,QI
      BMR=BMR+PI $ BMI=BMI+QI
19    CONTINUE
      BMTR(I)=BMR $ BMTI(I)=BMI
53    CONTINUE
```

```
      SUMR=0. $ SUMI=0.
      DO 80 I=1,N
      SUMR=SUMR+P(I)
80    SUMI=SUMI+Q(I)
      PUNCH 83
83    FORMAT(/25HTOTAL TRANSMISSION LOSSES//18X,4HREAL, 15X,4HIMAG/)
      PUNCH 84,SUMR,SUMI
84    FORMAT(10X,F15.8,6X,F15.8)
      PUNCH 85
      DO 86 I=1,N
86    PUNCH87,I,BMTR(I),BMTI(I)
      SUMR=0. $ SUMI=0.
      DO 90 I=1,N
      SUMR=SUMR+BMTR(I)
90    SUMI=SUMI+BMTI(I)
      PUNCH 203
      PUNCH 84,SUMR,SUMI
85    FORMAT(/8HMISMATCH//2X,1HI,18X,4HREAL,15X,4HIMAG/)
87    FORMAT(I3,10X,F15.8,6X,F15.8)
20    FORMAT(I2,2F20.8)
21    FORMAT(2F20.8)
22    FORMAT(F20.8)
23    FORMAT(19HNO DATA FOR BUS NO.,I3)
24    FORMAT(/19HNO. OF ITERATIONS =,I3//6HBUS NO,5H TYPE,6X,9HMAGNITUDE
     1,9X,5HANGLE,13X,1HP,15X,1HQ/)
25    FORMAT(I3,4(4X,F15.8))
26    FORMAT(/2X,1HI,6X,1HJ,17X,6HP(I,J),24X,6HQ(I,J)/)
27    FORMAT(I3,4X,I3,2(10X,F20.8))
200   FORMAT(I3,3X,5HSLACK,4(F16.8))
201   FORMAT(I3,3X,5HLOAD ,4(F16.8))
202   FORMAT(I3,3X,5HGEN. ,4(F16.8))
203   FORMAT(/14HTOTAL MISMATCH//18X,4HREAL,15X,4HIMAG/)
30    STOP
      END
      SUBROUTINE BRADMT
      DIMENSION YR(20,20),YI(20,20),ER(20),EI(20),ER1(20),EI1(20)
      DIMENSION IN(20),IK(3),JK(3),RK(3),XK(3),P(20),Q(20),VMAG(20)
      COMMON ER,EI,ERS,EIS,IN,VMAG,SUMR,SUMI,J,N,I,YR,YI,QI,PI
      COMMON  ERI,EII,IK,JK,RK,XK,ER1,EI1,P,Q
      DO 4 I=1,N
      DO 4 J=1,N
      IF(I-J)1,4,1
1     YRIJ=YR(I,J)
      YIIJ=YI(I,J)
      IF(YRIJ)3,2,3
2     IF(YIIJ)3,4,3
3     YR(I,J)=-YRIJ $ YI(I,J)=-YIIJ
4     CONTINUE
      RETURN
      END
      SUBROUTINE INPTRX
      DIMENSION YR(20,20),YI(20,20),ER(20),EI(20),ER1(20),EI1(20)
      DIMENSION IN(20),IK(3),JK(3),RK(3),XK(3),P(20),Q(20),VMAG(20)
      COMMON ER,EI,ERS,EIS,IN,VMAG,SUMR,SUMI,J,N,I,YR,YI,QI,PI
      COMMON  ERI,EII,IK,JK,RK,XK,ER1,EI1,P,Q
      DO1 I=1,N
      DO1 J=1,N
      YR(I,J)=0.
1     YI(I,J)=0.
      READ17, NCDRX
      NC=0
      IF(SENSE SWITCH 1)26,2
26    PUNCH 19
2     READ18, (IK(K),JK(K),RK(K),XK(K), K=1,2)
      NC=NC+1
      DO7K=1,2
      I=IK(K)
      J=JK(K)
      IF(I)3,7,3
3     IF(J)4,7,4
4     R=RK(K)
      X=XK(K)
      CALL AUPONB (1.,0.,R,X,G,B)
      IF(SENSE SWITCH 1)5,6
5     PUNCH 20, I,J,R,X,G,B
6     YR(I,J)=G
      YI(I,J)=B
      YR(J,I)=G
      YI(J,I)=B
7     CONTINUE
      IF(NC-NCDRX)2,8,8
8     IF(SENSE SWITCH 1 )25,27
25    PUNCH 21
27    DO 16 I=1,N
      SUMR=0.
      SUMI=0.
      DO 11 J=1,N
      YRIJ=YR(I,J)
      YIIJ=YI(I,J)
      IF(YRIJ)10,9,10
9     IF(YIIJ)10,11,10
10    SUMR=SUMR+YRIJ
      SUMI=SUMI+YIIJ
11    CONTINUE
      DO 16 J=1,N
      IF(I-J)12,13,12
12    YR(I,J)=-YR(I,J)
      YI(I,J)=-YI(I,J)
      GO TO 14
13    YR(I,J)=SUMR
      YI(I,J)=SUMI
14    IF(SENSE SWITCH 1)15,16
15    PUNCH22,I,J,YR(I,J),YI(I,J)
16    CONTINUE
      RETURN
17    FORMAT(I3)
18    FORMAT(2(2I2,2F18.8))
19    FORMAT(2X,1HI,6X,1HJ,7X,6HR(I,J),10X,6HX(I,J),10X,6HG(I,J),10X,
     1(I,J)/)
20    FORMAT(I3,4X,I3,4(3X,F13.8))
21    FORMAT(11HADMITTANCES/2X,1HI,6X,1HJ,19X,4HREAL,26X,4HIMAG/)
22    FORMAT(I3,4X,I3,2(10X,F20.8))
      END
      SUBROUTINE INPTGL
      DIMENSION YR(20,20),YI(20,20),ER(20),EI(20),ER1(20),EI1(20)
      DIMENSION IN(20),IK(3),JK(3),RK(3),XK(3),P(20),Q(20),VMAG(20)
      COMMON ER,EI,ERS,EIS,IN,VMAG,SUMR,SUMI,J,N,I,YR,YI,QI,PI
      COMMON  ERI,EII,IK,JK,RK,XK,ER1,EI1,P,Q
1     READ 8,INK,(JK(K),RK(K),XK(K),K=1,3)
      IF(INK)2,7,2
2     DO 6 K=1,3
      J=JK(K)
      IF(J)3,6,3
3     P(J)=RK(K)
      IN(J)=INK
      GO TO(4,5),INK
4     Q(J)=XK(K)
      GO TO6
5     VMAG(J)=XK(K)
6     CONTINUE
      GO TO 1
7     RETURN
8     FORMAT(I1,1X,3(I2,2F12.7))
      END
      SUBROUTINE INITAL
      DIMENSION YR(20,20),YI(20,20),ER(20),EI(20),ER1(20),EI1(20)
      DIMENSION IN(20),IK(3),JK(3),RK(3),XK(3),P(20),Q(20),VMAG(20)
      COMMON ER,EI,ERS,EIS,IN,VMAG,SUMR,SUMI,J,N,I,YR,YI,QI,PI
      COMMON  ERI,EII,IK,JK,RK,XK,ER1,EI1,P,Q
      ER(1)=ERS
      EI(1)=EIS
      DO4 I=2,N
      INI=IN(I)
      GO TO(1,2), INI
1     ER(I)=1.
      GO TO 3
2     ER(I)=VMAG(I)
3     EI(I)=0.
4     CONTINUE
      RETURN
      END
      SUBROUTINE SUMMAT
      DIMENSION YR(20,20),YI(20,20),ER(20),EI(20),ER1(20),EI1(20)
      DIMENSION IN(20),IK(3),JK(3),RK(3),XK(3),P(20),Q(20),VMAG(20)
      COMMON ER,EI,ERS,EIS,IN,VMAG,SUMR,SUMI,J,N,I,YR,YI,QI,PI
      COMMON  ERI,EII,IK,JK,RK,XK,ER1,EI1,P,Q
      SUMR=0.
      SUMI=0.
      DO 4 J=1,N
      IF(I-J)1,4,1
1     YRIJ=YR(I,J)
      IF(YRIJ)3,2,3
2     IF(YIIJ)3,4,3
3     CALL AMULTB (ER(J),EI(J),YRIJ,YIIJ,XR,XI)
      SUMR=SUMR+XR
      SUMI=SUMI+XI
4     CONTINUE
      RETURN
      END
```

```
SUBROUTINE VOLTAG
DIMENSION YR(20,20),YI(20,20),ER(20),EI(20),ER1(20),EI1(20)
DIMENSION IN(20),IK(3),JK(3),RK(3),XK(3),P(20),Q(20),VMAG(20)
COMMON ER,EI,ERS,EIS,IN,VMAG,SUMR,SUMI,J,N,I,YR,YI,QI,PI
COMMON  ERI,EII,IK,JK,RK,XK,ER1,EI1,P,Q
QI=-QI
ECI=-EII
CALL AUPONB (PI,QI,ERI,ECI,AR,AI)
AR=AR-SUMR
AI=AI-SUMI
CALL AUPONB (AR,AI,YR(I,I),YI(I,I),ERI,EII)
RETURN
END
SUBROUTINE GENERQ(QRI)
DIMENSION YR(20,20),YI(20,20),ER(20),EI(20),ER1(20),EI1(20)
DIMENSION IN(20),IK(3),JK(3),RK(3),XK(3),P(20),Q(20),VMAG(20)
ECI=-EII
CALL AMULTB (YR(I,I),YI(I,I),ERI,EII,CR,CI)
CR=SUMR+CR
CI=SUMI+CI
CALL AMULTB (CR,CI,ERI,ECI,QRI,QI)
QI=-QI
RETURN
END
```

APPENDIX - G

Generally, one is called upon to invert an admittance matrix using Node equation for the solution of Network problems.

Node equations using self and mutual admittances of a power system network can be solved by inverting the Admittance matrix, the analysis of which is as follows:

Using Kirchoff first law, it can be written that

$$I_{kk} = \sum_{j=1}^{b} I_{kj} \quad \ldots\ldots \quad (1)$$

where I_{kk} = total input current at node k

I_{kj} = branch current between node k and j

b = total number of branches at k

Also,

$$I_{kj} = (E_k - E_j) \; y_{kj} \qquad (2)$$

where E_k = voltage at node k

E_j = voltage at node j

y_{kj} = branch admittance between node k and j

Substituting equation 2 in equation 1 and combining,

$$I_{kk} = Y_{kk} E_k + \sum_{j=1}^{b} Y_{kj} E_j \qquad (3)$$

Where $Y_{kk} = \sum_{j=1}^{b} y_{kj}$ = self -admittance

$Y_{kj} = - y_{kj}$ = mutual admittance

solving equation 3 for voltage E_k

$$E_k = \frac{I_{kk} - \sum_{j=1}^{b} Y_{kj} E_j}{Y_{kk}} \qquad \ldots\ldots (4)$$

If k represents the driving point node and a current of 1 + j0 be impressed at k then,

$$E_{kD} = \frac{1.0 - \sum_{j=1}^{b} Y_{kj} E_j}{Y_{kk}} \qquad (5)$$

where E_{kd} = volt age at diagonal elements

E_j = voltage at adjacent nodes

For al lother node there is no current impressed i.e.,

I_{kk} = 0, therefore

$$E_{k\ OD} = \frac{-\sum_{j=1}^{b} Y_{kj} E_j}{Y_{kk}} \qquad (6)$$

Where E_{KOD} = voltage at off-diagonal elements Equation 5 is for diagonal element or node and equation 6 is for off-diagonal nodes. Solution of 5 and 6 is obtained by an iterative process with original assumptions of $E_j = 0$ At the end of iterative process when the volt age have converged to a desired precision, as desired by accuracy of results, the row of inverse of the original matrix has been computed.

Similarly, by energising all buses in turn the whole inverted matrix can be estimated easily.

APPENDIX - H

Variance Reduction Techniques

In a Monte Carlo Calculation the problem of round-off has little effect on the accuracy. The statistical deviation of result is more important.

A Monte carlo calculation consists of a series of independent trials at any one of which the standard deviation of a required parameter may be 6. The deviation of the estimate of such a parameter from N trials, that is of the arithmatic mean, of N observations, is then $\sigma N^{-\frac{1}{2}}$. Therefore an increase of one decimal figure in accuracy with which the parameter is estimated necessiates a hundred times increase in the number of trials.

It is obvious thus that if adequate precision can not be obtained with reasonable computer usage, some way must be found to reduce the errors other than an increase in machine time. And this requires the necessity of variance reducing techniques.

The actual discussion on variance corresponding to a particular distribution, the references (21, 30) provide good materials as regards matrix inversion by Monte Carlo method.

REFERENCES AND BIBLIOGRAPHY

1. J.T.Carlton and C.J. Baldwin
"Use of Compiler Programmes to solve Power System Problems on Large Computers, Trans.A.I.E.E.Vol 78, pp 1319 - 24, February, 1959

2. A.O.Thomas
Calculation of Transmission Line Impedances by Digital Computer, Vol. 78,pp 1270-3, Feb 1959

3. R.B. Shipley and D.W.Coleman
Power Circuit Representation for Digital Studies I.E.E.E. Vol 83, April, 1964

4. Bloodworth
Machine Printing of Flow Diagram from Digital Load flow solutions, I.E.E.E. Vol. 82, P 901-03, 1963

5. B.R.Myril, G.B.Reed, J.L.McMinley, H.K.Polk, R.V.Hergo, W.J.Martin.
A Digital Approach to Power System Engineering,- I, II, III and IV. Trans. A.I.E.E. Vol 80, June,1961

6. R.F.Cook, A.D.Patton
Using Digital Computer in Power System Design. Trans A.I.E.E.E., Vol. 79, p 223,June 1960

7. A.L.Toaleton
Digital solution of short circuit currents for Networks, Trans. A.I.E.E., Vol.,78, p 1720, Feb. 1960

8. H.E. Brown, C.E. Pearson, L.K. Kirchmayer and G.W.Stagg.
Digital calculations of Three phase short circuit by matrix method. Trans A.I.E.E. Vol 79 p 1277-82,1960

9. E.E.George
A new method of making Transmission loss Formula s Directly from Digital Power flow studies, Trans. A.I.E.E. Vol. 78, pp 1567-73, 1959

10. M.S. Dyrkacs, C.C.Young and F.J. Maginniss.
A Digital Transient stability Program Including
the effects of Regulator , Excitor and Governor
Response, Trans. A.I.E.E.,Vol. 79 pp 1245-57, 1960

11. F.M.Reza and S.Seely
Modern Network Analysis (book) McGraw Hill Book Co.
Inc. 1961

12. W.R.LePage
Analysis of Alternativing current circuits, (bbok)
McGraw Hill Book Co. Inc.,1952

13. Louis Weinberg
Network Analysis and Synthesis (book) Mcgraw Hill
Book Inc. 1962.

14. W.F.Tinney and C.M.McIntyre
A Digital method for obtaining a loop connection Matrix
Trans. A.I.E.E. Vol.79 pp 740-46, Oct-1960

15. N.Sato
Digital Calculation of Network Inverse and Mesh Transformation Matrices, Trans A.I.E.E. Vol., 79 pp 719-26
October 1960

16. H.W.Hale
A logic for identifying the trees of a Graph
Trans.A.I.E.E.Vol 80, p 195 June 1961

17. H.Edelmann
Numerical and Algebraic Generation of Mesh-Impedance
matrices by set-Theoritical Intersection on a Digital
Computor. I.E.E,E,Vol. 83,pp 397-402, April 1964

18. N.Sato and W.F.Tinney
Techniques for exploiting the sparsity of the Network
Admittance Matrix.
I.E.E.E Vol. 82, pp 944-9, Dec. 1963

19. W.F.Tinney
A new Approach to the solution of Power System Network Problems (Unpublished paper prepared at Bonneville Power Administration U.S.A. May 1966

20. V.Converti
Applying Row-by-Row Matrix Inversion to Power Systems Trans. A.I.E.E., Vol,78 pp 413-16, 1959

21. G.E.Forsythe and R.A.Leibler, Matrix Inversion by a Monte Carlo Method, MTAC Vol.4 pp 127-129, 1950

22. G.E.Forsythe and R.A.Leibler
Correction to the Article, Matrix Inversion by a Monte Carlo Process, MTAC Vol.5 pp 55, 1951

23. Herbert A.Meyer
Symposium on Monte Carlo Methods, John Wiley and Sons Inc. 1956

24. Dr. S.Das Gupta
Application of Monte Carlo Method for Analysis of D.C. Electrical Network ,Journal of the Institution of Engineers (India) Vol 43, No. 6 pt EL.3 pp 133, Eeb- 1963

25. Dr. S.Das Guptaand A.K,Roy
Application of Monte Carlo Method for the solution of D.C. Network from Node equations- Journal of the Institution of Engineers (India) Vol.44 No. 10 pt EL 5 June 1964

26. Dr. S.Das Gupta and A.K.Roy
Monte Carlo Games in the Analysis of A.C. Networks J.I.E (India) p 105 Feb, 1965

27. Stabitz and Larrivee
Mathematics and Computer, Mc Grawhill Book Company Inc,1957

28. Modern Computing Methods. Notes on Applied Science No. 16 1961 National Physical Laboratory, London.

29. Backenback
Modern Mathematics for the Engineer, McGraw Hill Book Co, Inc. p 336, 1956

30. Anthony Relston and Herbert S.Wilf
Mathematical Methods for digital computers
John Wiley & Sons p 78, Aug, 1964

31. P.M. Morse and G.E.Kimball
Methods of Operational Research. John Wiley & Sons Inc.1950

32. J.H.Wilkinson
Rounding Errors in Algebraic Processes. Notes on
Applied Sciamce No. 32 National Physical Laboratory
London, 1963

33. K.B.Misra
Monte Carlo methods and PossibleApplications to system
Problems, Research Journal, University of Roorkee,
Vol IX , 1966

34 D.N.Reps
Monte Carlo Simulation of Recidential Transformer
Loads, Trans A.I.E.E. Vol 81, pp 841-57 , Feb,1963

35. A.F. Glimn, R.Halermann, J.M.Henderson and
L.K.Kirchmayer
Digital Calculation of Network Impedances, Trans.A.I.E.E
pp 1285-95, Vol 74 Dec, 1955

36. R.B. Shipley
Discussion to the above paper (35) Ibid pp 1295
Dec.1955

37. H.W.Hale and J.B.Ward
Digital computation of Driving Point and Transfer
Impedances, Trans.A.I.E.E., Vol 76,pp 476-80 Aug.1957

38- L.A.Dunstan
Digital Load Flow studies, Trans.A.I.E.E., Vol.73,
p 825, 1954

39. J.M.Handerson
Automatic Digital computer solution of Load Flow studies
Trans.A.I.E.E. Vol. 73, p 1696, 1954

40. J.B.Ward and H.W. Hale
Digital Computer Solution of Power Flow Problems
Trans.A.I.E.E. Vol,75 p 398 ,June 1956

41. R.J.Brown and W.F.Tinney
Digital Solutions for Large Power Networks.,
Trans A.I,EE. Vol. 76, pp 347-55, June 1957

42. Donald McGill
Nodal Iterative Solution of Power Flow Problems
using IBM 604 Digital Computer, Trans.A.IE.E.
Vol. 76, pp 803-09, Oct. 1957

43. A.F.Glimn and G.W.Stagg
Automatic Calculation of Load Flows, Trans A.I.E.E.,
Vol 76, p 817, 1957

44. M.S.Dyrakacz, A.F.Glimn and D.G.Lewis
Extensions in Digital Load Flow Techniques using High
speed computer, conference Paper No. C.P 57-1056, 1957

45. R.H.Jordon
Rapidly converging Digital Load Flow. Trans A.I.E.E.,
Vol. 76, p 143, Feb 1958

46. M.S.Dyrkacz and F.J.Maginniss
A new Automatic Programme for Load Flow stuides
on IBM 704 Trans A.I.E.E., Vol. 78 pp 52-62, April 1959

47. J.E. Van Ness
Iterative Methods for Digital Load Flow studies
Trans. A.I.E.E., Vol. 78 pp 583-88 Aug.1959

48. H.W.Hale and R.W.Goodrich
Digital Computations of Power Flow- Some Aspects-
Trans. A.I.E.E., Vol 78,pp 919-24, Oct 1959

49. J.E.Wan Ness
Convergence of Iterative Load Flow studies,
Trans A.I.E.E., Vol. 78 pp 1590-97, Feb 1960

50. J.E.Van Ness
Conve.gence of Digital Load Flow studies
Vol. XXII Proceddings of the American Powe. Conference
1960

51. J.E. Van Ness
Elm.niation Method for Load Flow Studies
Trans.A.I.E.E., Vol. 80,pp 299-304, June 1961

52. H.E.Brown, G.K.Carter, H.H.Happ and C.E. Person
Power flow solution by Impedance Matrix
Iterative Method, I.E.E.E. Vol 82, April, 1963

53. J.H.Cronin and M.B. Newman
Digital Load Flow Programme for 1000 -bus system
I.E.E.E. Vol 83, p 718, July, 1964

MONTE CARLO METHODS AND POSSIBLE APPLICATIONS TO SYSTEM PROBLEMS

by

K.B. Misra[*]

duction

Two types of problems are met in practice. They are, Probabilistic and rministic according to whether or not they are directly concerned with the viour and out come of random processes.

Monte Carlo methods comprise that branch of experimental mathematics h is concerned with the experiments on random numbers.

In case of a probabilistic problem the simplest Monte Carlo approach is to rve random numbers, chosen in such a way that they directly simulate the ical random processes of the original problem, and to infer .the desired ion from the behaviour of these random numbers.

However, in case of deterministic problems also, once the theory has sed its underlying structure, one may perhaps recognise its structure associated some unrelated random process and thereby attack the problem.

f History

The name and the systematic development of Monte Carlo methods dates ut 1944. One of the earliest applications of the principle at the root of Monte o methods seems to have been made, as a mathematical recreation (in 1855), Mr. A. Smith of Scotland who found out the value of π in some 3204 trials as 53. Captain Fox got 3.1419 in 1120 trials. They threw a stick just enough each from one line to the other on a plane ruled by lines equally spaced and the probability that the stick will touch a line is $2/\pi$:

About 1950, matrix inversion and solution of partial differential equations

[*] Electrical Engineering Department, University of Roorkee, Roorkee.

by Monte Carlo methods were developed. And in the present paper it will be seen how these could be done by the method given by Forsythe and Leibler[1,2].

Two symposiums[3] were held at Los Angeles on June 27, 30 & July 1, 1949 and again on March 16, 17, 1954 at University of Florida, respectively. A series of papers were read and discussed and several applications of Monte Carlo nethods were suggested. Since then several scattered papers were published. In ast few years Monte Carlo methods have come back in to favour. This is due to etter recognition of those problems in which it is the best and sometimes the nly available technique.

Aonte Carlo Methods applied to system problems

In circuit problems one generally comes across a set of linear equations which could be solved to find the desired quantities in general. For example for ny D.C. network by properly selecting the linkset and hence corresponding tree r vice versa, the connection matrix C can be developed for any assumed reference ame. The loop resistances and loop voltages and currents could be written as

$$I_{branch} = C\, I_{loop}$$

$$I_{loop} = R_{loop}^{-1}\, e_{loop}$$

also $$R_{loop} = C^{t}\, R_{branch}\, C$$

Now here, the problem reduces to find the inverse of the matrix of R_{loop}. nce it is found, the solution to the network problem could easily be obtained. the network consists of a large number of branches and sources the problem of nding the inverse of a matrix poses a serious consideration and here we take help Monte Carlo methods and as shown by Das Gupta[4] that the solution to such mplicated networks could easily be had with fairly good values.

Similarly, the solution to the network problem could also be obtained if instead of mesh equations[5], one is called upon to use the node equations. There e solution would be obtained by finding the inverse of G matrix instead of R_{loop}. lso Enode and $E_{branches}$ would be calculated instead of I_{loop} and $I_{branches}$. So r solving a d.c. network has been discussed but little consideration would show it the A.C. network could also be solved on similar lines.[6]

The voltages and currents are related by the system of linear equations

$$Z_{11}\, I_1 + Z_{12}\, I_2 + \;\ldots\; + Z_{1n}\, I_n = E_1$$

One great advantage in favour of Monte Carlo method is that it inverts the matrix, row by row and thus sufficiently saves the labour in computations.

The main advantages of row by row inversion are

1. Since the calculations proceeds on a unit basis where unit is the bus or the node of the network. The matrix can be built up from this unit to the complete system.

2. There is complete freedom to operate on any row in any order.

The possibilities of applying the Monte Carlo method to the problem of Interference of power lines on communication lines or to find the line constants in case of an untransposed line having large number of conductors can also be explored. Monte Carlo method provides a simpler method of solution of Laplace equation[7]

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0$$ with given boundary conditions.

The physical situation for which it could arise may be an electron tube, high voltage electrical equipment or a magnetic field in a machine. In all these cases, one has to find the value of the electric or magnetic potential u_o at any point within the given boundary. The approximate equivalent of the above equation could be taken as

$$u_o = \frac{u_1 + u_2 + u_3 + u_4}{4}$$

Where u_1, u_2, u_3, u_4 are the values at the four corners around u_o.

Solution now depends on several random walks in the field starting with u_o point guided by random sequence, till the boundary is reached and the boundary value is the score of the game. Score on the basis of per game gives the potential at that point u_o. Thus Monte Carlo methods may find several applications in the field of Power System Engineering by proper simulation.

Comparative study of different methods for solving Linear Equations.

There are two types of methods[8] of solving a set of n Linear Equations in n unknowns (being large) viz,

Direct methods and Iterative methods. Gaussian Elimination is a direct method.

In the application of ordinary direct methods the number of multiplications is very large approximately $\frac{1}{3}n^3$ and the number of roundings correspondingly is great[12] thereby involving considerable error.

There arē iterative methods such as Seidel Gauss and simple or Jacobi iterations, best suited for sparse matrices (containing large proportions of zero elements) which are very common in network analysis.

According to G.E. Forsythe[9], conventional Cramer's rule procedure for solving n simultaneous equations in 'n' unknown requires evaluation of n+1 determinants of order n. As ordinarily defined, a determinant of order n is a sum of 1.2.3.4.. (n−1) n=n! different terms, each of which requires n−1 multiplications to obtain its value. If one were to proceed in this fashion, n+1 such determinants would require (n−1)! (n+1) ! multiplications to evaluate. For n=10, this is 359, 251, 200 multiplications to evaluate, requiring 114 years to get an answer at the rate of one multiplication each 10 sec. Even SWAC[9] (National Bureau of Standards Western Automatic Computer, a very fast digital computer) which could multiply 2600 times per second would need 38 hrs, for multiplications.

If the amount of work is measured by the number of multiplications required under the most favourable conditions, the Gauss Elimination method involves the amoun' of work α n^3 where n is the order of the matrix. The amount of work required by Monte Carlo method is given by an expression of the form n^2+n+b where b is fairly large. If no preliminary preparation aimed at reducing b is made then amount of work may be given by n+b.

The result of this varying dependence on the dimensionality of the problem is that the Monte Carlo method is theoritically the most efficient[3] than any other method for sufficiently large value of n. The value of n at which the method becomes more efficient than any other method for sufficiently large value of n, depends on the accuracy with which the solution is to be computed.

The Monte Carlo method may not give very high value of accuracy in the result by itself. Some iterations are necessary to give the desired accuracy.

The Monte Carlo method gives more accurate the off-diagonal terms than the leading diagonal terms such as aij and aji where i≠j gives nearly accurate values because generally with the circuit problems, one comes across a symmetric matrix only. This also cuts down the labour of finding the elements of the inverse matrix.

The accuracy which Das Gupta got for his problem[6] was fairly tolerable for most of the cases.

The computer programming for the Matrix inversion by Monte Carlo method[10] could be done and with the latest techniques available for variance reduction, the Monte Carlo method becomes a challange for the existing popular and effective method for solving a set of Linear equations with desired accuracy.

Generation of Random Numbers:—

When doing a Monte Carlo problem one focusses attention on three main points, viz;

1. Choosing the probability process
2. Generating sample values of the random variables on a given computing machine
3. Designing and using variance reducing technique.

The random numbers generated, thus play an important part in computations. There are several methods of generating random numbers (or pseudo-random).

The first suggestion of generating the random sequence came from Von Neumann who suggested that a random sequence could be generated by squaring a number, possibly, of more number of digits say for example 8 digit number and then taking the middle eights as the next number in sequence of random numbers. The retained eight digit number is once again squared and the middle eight digits be taken as the next in sequence of random numbers. This, however, shows unsatisfactory results if the number has less than eight digits and that the sequence develops unsatisfactory properties if extended beyond 700 or so eight-digit numbers in sequence.

Lehmer introduced an easy and more dependable method of generating random sequence. Das Gupta used the same sequence as described below for his computations in his first paper[4]. However he used different sequence in his second paper[5]. He used random sequence for that paper from the reference[11].

Lehmer's method calls for successive multiplications of an eight digit number by a suitable constant such as 23. For example, taking an eight digit number such as 12345678 and then multiplying this by 23, one gets a number as 283950594 which is a nine digit number. Making it 10 digit number by placing 0 on the exterme left and thus making the number as 0283950594. Now seperating out this number into two parts by removing 02 away from 83950594 and then substracting 02 from

83950594 to give 83950592. This is a second number in the sequence of random numbers. Again this new number is multiplied by 23 to give either 9 digit or 10 digit number in general. If it is a nine digit number there is necessity of placing a 0 on the extreme left to make it 10 digit number otherwise not.

Lehmer method provides a sequence which repeats itself but after 5882352 numbers have been computed, which is long enough for most of the applications. It contains almost 47 million random digits. Lehmer sequence could very well be computed on the computer by the relation.

$$Xn+1 = k\,Xn \pmod{M}$$

with $K=23$, $M=10^8+1$ for ENIAC[8]

In 1950, when several experiments[8] were performed for the solution of partial differential equations and inversion of matrices, the method followed for generation of random numbers was,

$$X_0 = 1,\ X_n + 1 = \rho\, X_n \pmod{2^{42}}$$

Where ρ is any odd power of 5. In practice $\rho = 5^{17}$ (the largest power of 5 accepted by the machine–SEAC) and X_0 could be any integer satisfying $X_0 = 1 \pmod 5$. This sequence has a period $2^{40} \sim 10^{12}$.

There are other methods[8,10] also for generating random numbers but above mentioned are only used generally.

Matrix Inversion by Monte Carlo method:—

The Monte Carlo method provides a simple computational approach to the statistical estimation of the elements of the inverse of a given matrix. The method will give any one element, a single row, or all of the elements of a matrix when certain conditions are met.

If B is a nxn matrix, its inverse is found by the method given below

$$\text{Let} \quad A = I - B, \tag{1}$$

where, I is a unit matrix (δ_{ij}).

The game which is described here is defined if

$$\max|\lambda_r(A^*)| < 1 \tag{2}$$

where $A^*_{ij} = |A_{ij}|$

If this is satisfied then

$$B^{-1} = (I-A)^{-1}$$
$$= 1 + A + A^2 + \dots \dots A^h + \dots$$
$$= \sum_{h=0}^{\infty} A^h$$

Therefore,

$$(B^{-1})_{ij} = \sum_{k=0}^{\infty} (A_k)_{ij}$$

A simple sufficient condition exists for testing whether condition (2) is satisfied.

$$\text{Let } S_j = \sum_{i=1}^{\infty} a_{ij} \tag{3}$$

If $S_j < 1$ for all j then $\max_i | \lambda_i (A) | < 1$

If matrix B is such that this test fails, it is possible in some cases to div:de B by a constant factor 'a' so that $B = aB'$ and using $I-B'=A'$,

$$\max | \lambda_i (A') | < 1 \tag{4}$$

Then $(B')^{-1}$ is obtained by the Monte Carlo method

and $$[B_{ij}]^{-1} = \frac{1}{a}[B'_{ij}]^{-1} \tag{5}$$

Now a probability model is to be developed for this game. It is assumed that sum of all the elements in any row of A^* [$= a_{ij}^* = | a_{ij} | = | \delta_{ij} - b_{ij} |$] is less than unity ie,

$$\sum_{j=1}^{n} a_{ij}^* < 1 \tag{6}$$

But if this is not possible then dividing by a factor 'a', may fulfil this condition.

The Monte Carlo method for obtaining the inverse of matrix B is to play a game G_{ij}, for a large number of times and the expected payment of the game is C, ie $(B)_{ij}^{-1}$. If N is the number of times the game is being played then according to the result of Kolmogorff on the strong law of large numbers, we get

$$\lim_{N\to\infty} E(G) = C \tag{7}$$

Now to find inverse of B, first matrix A is found by the relation $I-B=A$.

The parameter A^* is the same as A but has the elements with magnitudes only. The value factor matrix, V, is then developed the entries of which are either $+1$ or -1 according to a_{ij} being positive or negative

therefore, $$a_{ij} = a_{ij}^* \, v_{ij} \tag{8}$$

where v_{ij} is the element in the value factor matrix. Equation (8) could be compared with that the given by Forsythe and Liebler[1], ie.

$$a_{ij} = P_{ij} \, v_{ij} \tag{9}$$

where P_{ij} is the element of the probability model P. Then from (8) & (9) we get,

$$p_{ij} = a_{ij}^* \tag{10}$$

It can be shown that stop probability model p_i for any i th row is given by[1]

$$p_i = 1 - \sum_{j=1}^{n} p_{ij} \tag{11}$$

The possible probability model is given below

P_{11}	P_{12}		P_{1n}	P_1
P_{21}	P_{22}		P_{2n}	P_2
P_{n1}	P_{n2}		P_{nn}	P_n

This may be noted here that probability model is a $(nxn+1)$ matrix.

The P matrix is then multiplied by M a large arbitary number, and each element of the matrix after multiplication, is rounded off to the nearest integral value. The error in the rounding off decreases as the value of M chosen increases but final result is independent of M. Now the new entries in the probability model would be

n_{11}	n_{12}		n_{1n}	n_1
n_{21}	n_{22}		n_{2n}	n_2
...	...	...	...	...
n_{n1}	n_{n2}		n_{nn}	n_n

where $n_{11} = r_{11}^{M}$

$n_1 = p_1^{M}$

and so on.

here $n_{11} + n_{12} + n_{13} \ldots + n_1 = M$.

Next the field numbers are allotted to each element of the probability model row-wise as shown below

row 1

n_{11}	$0 \to (n_{11} - 1)$
n_{12}	$n_{11} \to (n_{12} - 1)$
	......
n_1	$(n_{11} + n_{12} + \ldots n_{1n}) \to M\text{-}1$

row 2

n_{21}	$0 \to n_{21}$
n_{22}	$n_{21} \to (n_{22} - 1)$
...	......
n_2	$(n_{21} + n_{22} + \ldots n_{2n}) \to M-1$

Rules of the game to obtain the ith row of the inverse

(1) From the set of random numbers, first number is drawn and this number is checked in the field deck corresponding to ith row and the random number is found to lie in a particular field and the entry n_{ij} in the field is located the number j is further tested for $j \doteq n + 1$

(2) If $j = n + 1$ ie $n_{ij} = n_i\,(n + 1)$ the game should be stopped and a payment of $+\, n_i^{-1}$ is made in favour of ith entry of the ith, row

(3) If $j \neq n + 1$, the game does not stop and since the last entry in which the random number fell was n_{ij}, Now looking up for the next random number in the jth row to locate a the random number in ^{n}jk field. Again this k is test for $k = n + 1$.

(4) Again if $k = n + 1$, the game should be stopped and a score of $\pm\,(n_i)^{-1}$ is made in favour of the jth entry of i th row.

This must be noted here that although help of other rows of the matrix to be inverted is taken entries of the score would be made in favour of a particular row which is being inverted.

(5) Once more if $k \neq n + 1$, the game is not stopped and for the next random number in sequence, the field in kth row in which the number lies is

looked up and so on. The same procedure for locating the field for the random numbers in sequence proceeds on till the stop probobility field is hit and the score be recorded against the particular element of the ith row for which the game was played. It may be here once again made clear that the score for a particular element would be + or − depending upon the route followed in the game. Suppose that the game proceeds on from the deck 1 to deck 3 and once again to deck 2 and then stop probability field is hit. Then the exact route followed is (1)→(3)→(2), hence the score would be + or − depending upon the elements of the value factor matrix

The exact value of the score made against the particular element of row to be inverted would be given by

$$= v_{13}\ v_{32} \cdot 1$$

where v_{13} and v_{32} are the elements of the value factor matrix set up already. This is the big advantage as mentioned earlier of Monte Carlo method, that there is row inversion of the matrix and thus facilitate work and unnecessary or un-wanted rows or elements may be ommitted for inversion. The game is played as per conditions for a large number of times and score be noted as above. The average score on a per game basis gives the solution.

Conclusions

Monte Carlo Games provide a very good method for s lving Linear equations. The method is good for an approximate solution of the equations but with the latest variance reducing technique suitable for high speed computer the method becomes quite promisir.g. The solution converges rapidly in the begining but is refined only for further large number o(games. Also, if this method is used in conjuction with Some iteration, the method may cut short the computation time and error.

Acknowledgements:

The author is thankful to Dr. P. Mukhopadhyaya for initiating the topic and guidance.

Thanks are due to Prof. C.S Ghosh for the permission to publish and to Mr. M.N. Keshava Rao of S.E.R.C., Roorkee for his help in preparing the programme.

REFERENCES

1. G.E. Forsythe and R.A. Leibler, Matrix Inversion by a Monte Carlo Method,

MTAC, Vol. 4, 1950, pp. 127–129.

2. G.E. Forsythe and R A. Leibler, Correction to the Article, "Matrix Inversion by a Monte Carlo Process," MTAC, Vol. 5, 1951, p. 55.

3. Herbert A. Meyer "Symposium on Monte Carlo Methods", John Wiley and Sons, Inc. 1956.

4. Dr. S. Das Gupta, "Application of Monte Carlo Method for Analysis of D.C. Electrical Networks", Journal of the Institution of Engineers (India), Vol. 43, No. 6, pt. EL3, Eeb. 1963, pp. 133.

5. Dr. S. Das Gupta and A.K. Roy, "Application of Monte Carlo Method for the solution of D.C. Network from Node Equations, Journal of the Institution of Engineers (India) Vol 44, no. 10, pt. EL 5, June 1964.

6. Dr. S. Das Guta & A K. Roy. "Monte Carlo Games in the Analysis of A.C. Networks", J.I.E. (India) Feb. 1965, p. 105.

7. Stibitz and Larrivee, "Mathematics and Computer, Mc Grawhill Book Company Inc., 1957.

8. Modern Computing Methods, Notes on Applied Science No. 16. 1961. National Physical Laboratory, London.

9. Backenback "Modern Mathematics for the Engineer", McGraw Hill Book Company Inc., 1956, p. 336.

10. Anthony Relston and Herbert S. Wilf, 'Mathematical Methods for digital computers', John Wiley & Sons, Aug. 1964, p. 78.

11. P.M. Morse and G.E. Kimball. "Methods of Operational Research", John Wiley and Sons Inc., 1951.

12. J.H. Wilkinson 'Rounding Errors in Algebraic Processes', Notes on Applied Science No. 32, National Physical Laboratory, London, 1963.

Appendix

a. PROGRAMME:—The programme is based on the method given in reference 7.

```
C   C MONTE CARLO METHOD FOR SOLUTION OF EQUATIONS— Z
    COMMON N
    DIMENSION B (30), NP (25), A (30, 31), MAT (30, 30)
    READ 1, N GAME, N SIZE, NNO  $ N1 = NNO + 1
```

```
     G = N GAME  $ G = 1./G
     READ 1, (NP (I), I = 1, NNO)  $ NP (N1) = N GAME + 1
 1   FORMAT (25I3) $ N MORE = N SIZE + 1$ READ 2, ((A (I, J), J = 1,
                                                   N MORE), I = 1, N SIZE)
 2   FORMAT (4E16.8)
     SKEL = 99999.$ N SKEL = SKEL$ DO 3I = 1, N SIZE
     DO 4 J=1, N SIZE$ M FIXD=A (I, J)* SKEL
     IF (J-1) 21, 21, 22
21   MAT (I, 1) = M FIXD $ GO TO 4
22   MAT (I, J) = MAT (I, J-1) + M FIXD
 4   CONTINUE
     M STEP = N SKEL-MAT (I, N SIZE)
     IF (M STEP) 8, 8, 14
 8   PRINT 9, I
 9   FORMAT (I3, 19H ROW  NOT NORMALIZED)
     STOP
14   STEP = M STEP
 3       B (I) = A (I, N MORE)* SKEL/STEP
     IROW = 1
18   SKO = 0 $ NO = 1
     DO 15 N CONT = 1, N GAME
     IF (N CONT-NP (NO))'100, 101, 100
101  NO = NO + 1
     G1 = N CONT $  SKOR = SKO/G1
     PUNCH 16, NCONT, IROW, SKOR
16   FORMAT (I3, I3, E16.8)
100  I = IROW
12   CALL RANDOM
     Do 7 J = 1, N SIZE
     JEND = N SIZE + 1-J
     IF (N-MAT (I, JEND)) 7,6,6
 7  CONTINUE
    I = 1 $ GO TO 12
 6  If (JEND-N SIZE) 11, 15, 15
11  I = JEND + 1 $ GO TO 12
15  SKO = SKO + B (I) $ SKO = SKO* G $ PUNCH 19, IROW, SKO
19  FORMAT (1XI3, 10XE16.8)
    IROW = IROW + 1
    If (IROW — N MORE) 18, 17, 17
17  STOP
    END
```

b. SUBROUTINE RANDOM:—Could be written on the basis of the methods already mentioned or from references 7,10 mentioned herein.
Subroutine should necessarily be in machine language.

c. The problem given in reference 7 was tried, the results were found to be quite satisfactory.