"Circuit design is a creative activity that requires not only experience and skill but intuition and luck as well " Gray,P.E. and Searle, C.L. in ' Electronic Principles - Physics, Models, and Circuits '

"I must create a system or be enslaved by another man's"
Blake

"The reasonable man adapts himself to the world; the unreasonable one persists in trying to adapt the world to himself. Therefore all progress depends on the unreasonable man "
Shaw

THESIS APPROVAL

(i) INTERNAL EXAMINER

B.J. Olufeagba (Prof.),
Dept. of Electrical Engineering,
University of Ilorin,
Ilorin,
Nigeria

.....

.....

(ii) INTERNAL EXAMINER

Dr. R.K. Gupta,
Dept. of Electrical Engineering,
Ahmadu Bello University,
Samaru - Zaria,
Nigeria

.....

.....

(iii) EXTERNAL EXAMINER

Dr. L.A. Buraimoh-Igbo,
Dept. of Electronics &
Electrical Engineering,
University of Ife,
Ile - Ife,
Nigeria

.....

.....

APPENDIX I

INTERACTIVE D.C. HODS COMPUTER AIDED DESIGN OF ELECTRONIC NETWORKS

BY

SODE-SHINNI NMADU JONATHAN RUMALA

A thesis submitted to the postgraduate school, Ahmadu Bello University; in partial fulfillment of the requirements for the degree of M.SC., CONTROL AND INDUSTRIAL ELECTRONICS.

Department of Electrical Engineering,
Faculty of Engineering,
Ahmadu Bello University,
Zaria

OCTOBER, 1981.

APPENDIX II

I hereby declare this thesis has been composed by myself and that it is a record of my own research work.

It has not been accepted in any previous application for a higher degree. All quotations are distinguished by quotation marks and sources of information are specifically acknowledged by means of references.

SODE-SHINNI MUADU BELLA

October 1981

Acknowledgements XVII

PREFACE XX

CHAPTER 1: SURVEY OF MATRIX METHODS FOR ELECTRICAL NETWORK ANALYSIS 1

Introduction 1

1.1 Definitions 1

1.2 The standard branch 2

1.3 Network tree generation 6

1.3.1 The maxima tree branch per node algorithm 6

1.4 A survey of some matrix network analysis methods 13

1.4.1 Cutset based analysis 13

1.4.2 Tieset, (loop), matrix based analysis 15

1.4.3 Mesh matrix based analysis 17

1.4.4 Nodal matrix based analysis 20

1.5 Analysis via the nodal matrix method - reasons for choice 22

CHAPTER 2: THE NEWTON-RAPHSON ITERATIVE APPROACH TO NONLINEAR CIRCUIT ENUMERATION - A SURVEY 24

Introduction 24

2.1 The modified Newton-Raphson procedure applied to nodal circuit analysis 25

2.2 Convergence problems of the Newton Raphson iterative procedure and methods used in the thesis to overcome them 27

CHAPTER 3: ITERATIVE DEVICE MODELLING FOR SOME NONLINEAR RESISTIVE DEVICES 37

Introduction 37

3.1 The iterative diode model 39

The Ebers-Moll bipolar transistor model 39

The iterative Mosfet transistor model 41

CHAPTER 4: CONSTRUCTION AND FORMULATION OF THE GENERAL D.C. MODE ELECTRICAL NETWORK ANALYSIS PROGRAM 45

Introduction 45

4.1 Atrributes of the program 46

4.2.1. The main program 48

4.2.2 The diode modelling routine 48

4.2.3 The transistor modelling routine 48

4.2.4 The change element value routine 53

4.2.5 The dnagling node check routine 53

4.2.6 The iterative voltage boundary routine 53

4.3 Sequential analysis of the interactive nature of the program 57
4.4 File assignments 62
CHAPTER 5: APPLICATIONS OF THE ELECTRONIC CIRCUIT ANALYSIS PROGRAM-D.C. MODE ELECTRONIC CIRCUITS 63
Introduction 63
5.1 Typical iterative input sequence - analysis of a voltage regulator 66
5.2 Linear resistive network dependent source-Op. AMP., transistor combination 70
5.3 Multi-stage circuit analysis-the source mobility problem 73
5.4 Variation of iterative convergence with magnitude of feed-back for a single transistor amplifier 76
5.5 The temperature variation routine 76
5.6 Analysis of a standard voltage regulator - circuit with active component in the feedback loop 80
5.7 Incremental transistor models and small signal voltage gain calculations 82
5.8 The cut-off transistor anomaly - analysis of a voltage regulator incorporating a current limmitter 84

5.9 Comparison of the thesis program with some of the most well-known D.C. mode network analysis programs 87

CHAPTER 6: CONCLUSION 89

6:1 Areas of further research 91

APPENDICES

APPENDIX I Title III

APPENDIX II Declaration IV

APPENDIX III Program listings of results of some iterative analysis 94

APPENDIX IV Construct of the 'D.C. mode computer aided design of electronic networks' program 116

APPENDIX V References 120

LIST OF FIGURES

Fig.1.1.1 An undirected network graph 3

Fig.1.1.2 Tree formation for the network graph of Fig.1.1.1 3

Fig.1.1.3 Loop formation for the graph of Fig.1.1.1 3

Fig.1.1.4 Link formation for the graph of Fig.1.1.1 3

Fig.1.2 The standard branch 5

Fig.1.3 A state diagram for the maxima tree-branch per node algorithm 8

Fig.1.4 Network graph used to illustrate the applications of the maxima tree-branch per node algorithm 11

Fig.1.5 Network tree graph for the graph of Fig.1.4 via the maxima tree branch per node algorithm 12

Fig.1.6 Choice of basic cuts for the graph of Fig.1.1.1 14

Fig.1.7 Choice of loops for the graph of Fig.1.1.1 16

Fig.1.8 Formation of meshes for the graph of Fig.1.1.1 18

Fig.1.9 Non-planar network graph to illustrate the non-applicability of mesh matrix method in the general to non-planar networks 19

Fig.1.10 Choice of current directions for nodal analysis withrespect to Fig.1.1.1 21

Fig.2.1 Illustration of oscillatory convergence with direction of iteration 28

Fig.2.2 Illustration of oscillatory divergence with direction of iteration 29

Fig.2.3 The use of bounding to ensure convergence in an otherwise divergent iteration 30

Fig.2.4. Application of mean value theorem to diode characteristic in the derivation of convergence criterion 32

Fig.2.5 Choice of iterative voltage values to enhance convergence 35

Fig.3.1 Non-linear device characterisation 38

Fig.3.2 The iterative diode model 40

Fig.3.3.Static form of the Ebers-Moll npn transistor model 42

Fig.3.4 The iterative model for the npn transistor 42

Fig.3.5 An equivalent Ebers-Moll model for the MOSFET 43

Fig.3.6 The iterative equivalent for the enhancement MOSFET 43

Fig.4.1 Program flow-chart for the general D.C. mode electronic circuit analysis program 49

Fig.4.2 Flow-chart for the diode modelling routine 51

Fig.4.3 Flow-chart for the transistor modelling routine 52

Fig.4.4 Flow-chart for the dangling node check routine 54

Fig.4.5 Flow-chart for the change element value routine 55

Fig.4.6 Flow-chart for the iterative voltage boundary routine 56

Fig.5.1 Simple series voltage regulator - circuit incorporating zener diode 67

Fig.5.2 OP. AMP. model illustrating dependent sources 72

Fig.5.3 OP.AMP.-transistor combination - circuit incorporating linear dependent source 72

Fig.5.4 Two stage amplifier - source mobility illustration 75

Fig.5.5 Source mobility consideration withrespect to circuit of Fig.5.4 75

Fig.5.6 Circuit with linear resistive feedback - demonstration of stability of iteration convergence with magnitude of feedback 77

Fig.5.7 Circuit used to test the temperature variation routine 79

Fig.5.8 Circuit with non-linear transistor feedback - a voltage regulator 81

Fig.5.9 The voltage regulator of Fig.5.8 re-drawn for iterative analysis 81

Fig.5.10 Circuit used to illustrate small signal voltage gain calculations via the pi transistor model 83

Fig.5.11 Pi-model representation of circuit of Fig.5.10 83

Fig.5.12 Voltage regulator incorporating a current limmitter - illustration of cut-off transistor anomaly 86

Fig.5.13 The circuit of Fig.5.12 redrawn for circuit analysis 86

APPENDIX FIGURES - COMPUTER PRINT-OUTS

Fig.A1 Simple voltage regulator design data 95

Fig.A2 OP.AMP.-transistor combination analysis data 97

Fig.A3 Two stage amplifier - source mobility illustration data 101

Fig.A4 Effect of feed-back on convergence data 105

Fig.A5 Data of a pnp transistor amplifier operated at 350°K 108

Fig. A6 Data of a voltage regulator design - circuit incorporating non-linear element in the feed-back loop 111

Fig. A7 Listing of main and subroutine programs 117

LIST OF TABLES

Table 1.1 Incidence matrix for graph of Fig.1.4 3

Table 5.1 Table of node voltages for the simple voltage regulator of Fig.5.1 70

Table 5.2 Node voltages for the linear dependent source-transistor combination example 71

Table 5.3 Node voltages for the circuit of Fig.5.5 used to illustrate source mobility 74

Table 5.4 Variation of node voltages with temperature for circuit of Fig.5.7 78

Table 5.5 Node voltages variations with load for circuit of Fig.5.13 85

Table 5.6 Node voltages for circuit of Fig.5.11 82

Table5.7 Node voltage variations with load for circuit of Fig.5.13 85

Table 5.8 Comparison of the thesis functions with other well-known ECAP programs 88

ACKNOWLEDGEMENTS

The inspirations gained from my tutors at the various stages of my schooling is hereby acknowledged:-

(i) Prof. B.J.Olufeagba, the course convener, and Dr.R.K.Gupta; who supervised this thesis and offered useful advice, suggestions, criticisms, and test cases, in order to make the thesis a success.

(ii) Prof. Buba Bajoga, the Head of Department of Electrical Engineering, Ahmadu Bello University, as well as the staff of the Department of Electrical Engineering, Ahmadu Bello University, who lectured me during the course.

(iii) Dr. Ranjangan, my first year Physics tutor at the University.

(iv) Derek Joy, my Secondary school Form master, who wrote as an annual report"..... a real thinker with a great thirst for knowledge".

S.N.J.Rumala

PREFACE

The subject of this thesis is to design a general purpose D.C. mode network analysis program to perform the D.C. initial condition analysis of discrete component semiconductor circuits with special reference to bipolar transistors, and review various matrix methods of achieving the same.

The maxima tree branch per node algorithm is an original formulation.
The program, written in Fortran, is truly conversational interactive, and the cunny way this is achieved is original and has not been seen used in any published Fortran program - viz; the interactive nature is maintained no matter which output medium the program results are to be assigned.

The program aims at achieving the goals laid out by Kuo[11,12], as the fundamental criteria for any ideal computer aided circuit design program, viz;

(i) automatic parameter modification feature to enable on line breadboarding;

(ii) error check routine to enable the user asses the reliability of the output;

(iii) a convenient and simple language to be used to describe circuit topology so that even those with no engineering or even programming experience could be able to feed-in circuit information from a given circuit schematic;
(iv) it should be possible for the program to accomodate a wide class of models without drastic change in the main program construct.

It is appreciated after Magnus and Walsh[12] that a"non-linear program require more programming experience than a linear program", and that " under certain conditions, a compiler will generate incorrect coding for the Fortran program".
The program has been tested over a very wide range of circuits, and have given correct solution for circuits for which the iteration would converge.

For most non-linear D.C. condition networks, a solution is arrived at within nine iteration steps, if atall a solution would be arrived at; an indication of the very fast convergence of the scheme as compared to ordinary Newton-Raphson schemes.

Sode-shinni N. Rumala(B.SC,D.E.S)
October 1981.

CHAPTER ONE

SURVEY OF MATRIX METHODS FOR ELECTRICAL NETWORK ANALYSIS

INTRODUCTION

Electrical networks may be decomposed into two classes:- linear networks and nonlinear networks. The analysis of such networks may be broken down into two parts:-

(i) equation formulation

(ii) numerical solution.

The formulation of circuit equation for computer solution, except in trivial ones, employ various matrix methods. The formulation of circuit matrices is further enhanced through an understanding of the topological properties of the network via network graph.

The most common matrix techniques toward network solution are examined, and reasons for the choice of the nodal matrix analysis technique in this thesis is given.

1.1 DEFINITIONS

(a) A tree is a set of all branch connections of a network graph such that no loops are contained therein.

(b) A loop is any closed path in a network graph.

(c) A link is any branch of a network graph that does not form part of a specified tree for the network.

(d) A standard branch is a connection of at least one of the following elements between a specified pair of nodes :- (i) current source (ii) voltage source (iii) admittance or impedance.

For a graphical illustration of the above definitions, the graph of Fig1.1.1 is considered.

A network tree for the graph of Fig.1.1.1 is given in Fig.1.1.2. Fig.1.1.3 illustrates the formation of loops for the network graph. The links for the tree of Fig.1.1.2 is given in Fig.1.1.4.

1.2 THE STANDARD BRANCH[1]

The requirements of the analytical methods to be described in this thesis can be met via a mathematical formulation of the standard branch.

Let:-

r = rth branch of a circuit

Y_{rr} = self admittance of the rth branch

I_r = fixed current source across the rth branch.

Then the standard branch has as its elements, Z_{rr}, Y_{rr}, I_r, and V_r. For link current(tieset) analysis, the fixed current source is assumed to be between terminals(nodes) that define the branch, while for tree branch(cutset) analysis, the fixed voltage source

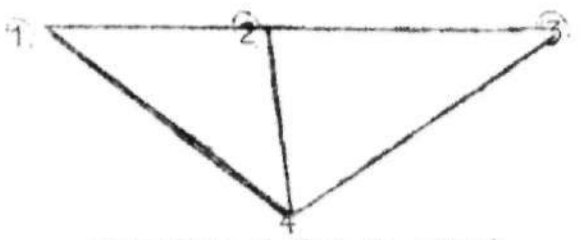


Fig.1.1.1. An undirected network graph.

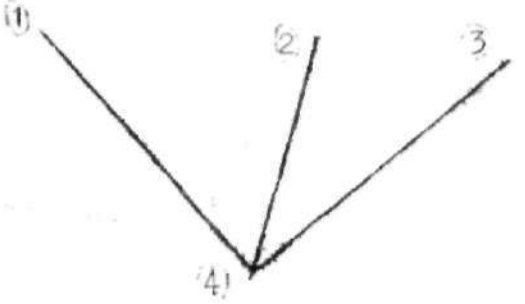


Fig.1.1.2 .Tree formation for the network of Fig.1.1.1

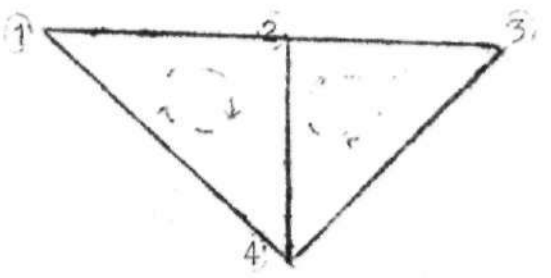


Fig.1.1.3. Loop formation for the graph of Fig.1.1.1
(loops represented by arrowed dashes).

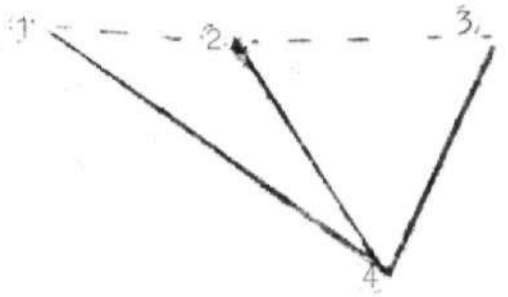


Fig.1.1.4. Link(dashed) formation for the graph of
Fig.1.1.1.

is placed in series with impedance, Z_{rr}, of the rth branch.

Fig.1.2 is a standard branch illustration.

For the rth branch representation, we have,

$$V_r = Z_{rr}\Psi_r \quad \text{.....(Ohmns law)}$$

For all branches simultaneously,

$$\underline{V} = \underline{Z}\,\underline{\Psi} \quad \text{.....(i)}$$

Now,

$$\underline{\Psi} = \underline{YV} \quad \text{.....(ii)}$$

Simple substitution yields

$$(\underline{E} - \underline{ZI}) + \underline{e} = \underline{Z}\,\underline{i} \quad \text{.....(iii)}$$

$$(\underline{I} - \underline{YE}) + \underline{i} = \underline{Y}\,\underline{e} \quad \text{.....(iv)}$$

Now,

$\underline{A}^t\,\underline{i} = \underline{0}$, where the superscipt, t, represents transpose.(Kirchoff's current law)

where $\underline{A}$ is the reduced incidence matrix.

Assume a transformation

$$\underline{A}\,\underline{e}' = \underline{e} \quad \text{.....(v)}$$

then

$$\underline{A}^t(\underline{I} - \underline{Y}\,\underline{E}) = \underline{A}^t\,\underline{Y}\,\underline{A}\,\underline{e}' \quad \text{.....(vi)}$$

Hence,

$$\underline{e}' = (\underline{A}^t\underline{Y}\,\underline{A})^{-1}(\underline{A}^t(\underline{I} - \underline{YE})) \quad \text{.....(vii)}$$

with the nomenclature $(\ )^{-1}$ implying the inverse of the expression in angular brackets, where

$\underline{e} = \underline{A}\,\underline{e}'$ = branch voltages

$\underline{i} = \underline{Y}\,\underline{e} - (\underline{I} - \underline{Y}\,\underline{E})$ = branch currents

$\underline{v} = \underline{e} + \underline{E}$ = element voltages.

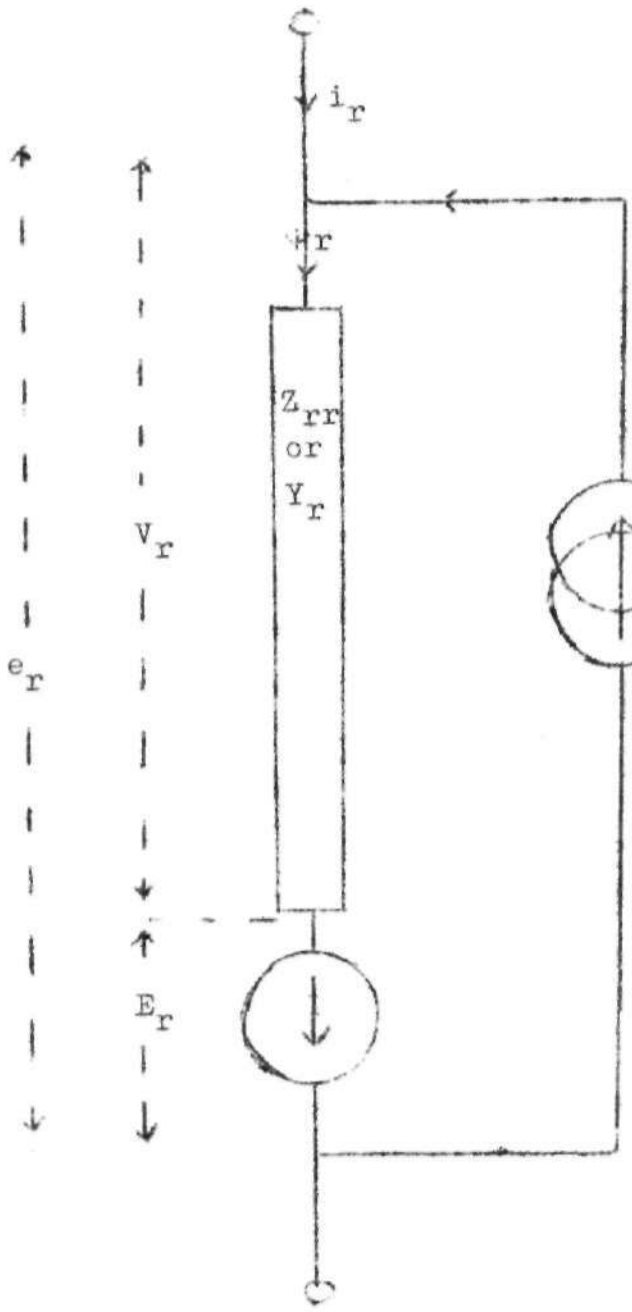


Fig.1.2 : The standard branch

1.3 NETWORK TREE GENERATION

Most analytical methods used in circuit enumeration depends on the generation of a specific tree for the network concerned and so are the design of printed circuit boards, road networks, optimal time slot assignments, system design, to mention but a few.

1.3.1 THE MAXIMA TREE BRANCH PER NODE ALGORITHN

The maxima tree branch per node algorithn has possible applications in areas where there is priority ordering of paths and/or nodes.

If we define a terminal node as a node that has been used to form a branch of a tree subgraph, and a starting node as the node from which new tree branches are being generated; and make similar definitions for the starting branch, the maxima tree branch per node algorithn may be formulated as follws:-

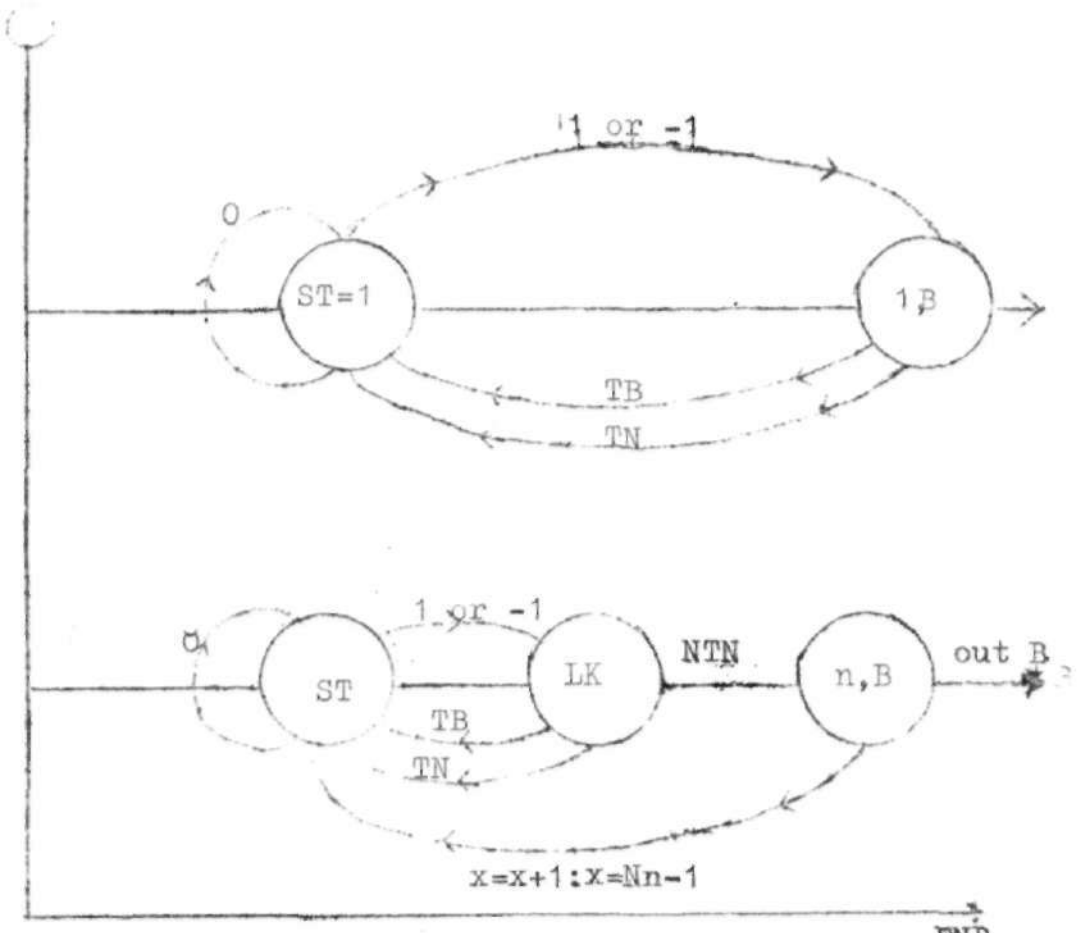
Form the incidence matrix.

(i) Use node 1 as the starting node. Scan withrespect to node 1 all branches. If a 1 or -1 is found in the incidence matrix at say, branch B, put the pir node-branch (1,B) into the look-up table. For each branch so found, scan the nodes attached to the branch until a 1 or -1 is found at say, node n. Enter into the look-up table, the pair (n,B).

Print out B as a successful tree branch. Continue until all branches connected to node 1 are exhausted. Use node 2 as the next starting node.
(ii) Scan withrespect to the starting node, n, all branches attached to n until a 1 or -1 is found at branch,B, say. Check the look-up table for the presence of B. If B's presence is encountered in the look-up table, discard branch B as an unsuitable candidate for the branch of the tree subgraph. If B is not a terminal branch, scan withrespect to branch B all nodes until a 1 or -1 is found at node n1. Check if node n1 is in the look-up table. Print out B as a successful branch of the tree subgraph.
Continue for all branches attached to the starting node. Continue for all nodes.
A state diagram for the maxima tree branch per node algorithn is presented in Fig 1.3.

Consider the algorithn applied to graph of Fig1.4. A branch - node matrix representing the incidence matrix is constructed by representing the (fro-node, branch) element by +1, and the (to-node, branch) element by -1. This gives the incidence matrix representation of the graph of Fig.1.4 as in Table 1.1.
Proceeding to apply the algorithn yields :-

Maxima of tree branches at node 1 = Branch 1,Branch2
Maxima of tree branches at node2 = Branch8,Branch9, Branch 10
Maxima of tree branches at node 3 = no Branch
Maxima of tree branches at node 4 = Branch 4.



KEY:-

ST= starting node
NTB = non-terminal branch
Nn = total number of circuit nodes
LK = look-up table
TB = terminal branch
n,B = node and branch added to the look-up table
x= number of tree branches already generated
NTN = non-terminal node
TN = terminal node.

Fig.1.3. A state diagram for the maximal tree-branch per node algorithn.

	BRANCH										
NODE	1	2	3	4	5	6	7	8	9	10	11
1	1	1	0	0	0	0	0	0	0	0	0
2	-1	0	0	0	0	0	0	1	1	-1	0
3	0	0	-1	0	0	0	0	-1	0	0	11
4	0	-1	1	-1	1	0	0	0	0	0	0
5	0	0	0	1	0	1	-1	0	0	0	0
6	0	0	0	0	0	0	1	0	-1	0	-1
7	0	0	0	0	-1	-1	0	0	0	1	0

TABLE 1.1 Incidence matrix of graph of Fig1.4

For the case of graph of Fig 1.4, the number of possible tree branches stop at six.

The procedure of application of the algorithm to the graph of Fig 1.4 proceeds as follows:-

(i) Starting at node 1, branches 1 and 2 are taken as tree branches. The node-branch pairs(1,2) and (1,4) are stored in the look-up table.

Scan goes to node 2. Branches 8, 9,and 10 are taken as tree branches. Branch 1 cannot be taken again as forming a new tree branch because node2 is a terminal node already stored in the look-up table. For branch8, a downward vertical scan yields a -1 at node 3. Node 3 had not previously been recorded as terminal node in the look-up table. Branch 8 is therefore recorded as a branch of the look-up table. A similar procedure records branches 9 and 10 as tree branches ,

and node branch pairs(2,8),(6,9),(2,10),and (7,10) into the look-up table.

With node 3 as the starting node, a horizontal scan of the incidence matrix along node 3 yields a -1 at each of the branches 3, and 8, and a 1 at branch 11. Branches 3,8, and 10 are terminal branches, and are therefore discarded as new candidates for the tree subgraph branch generation.

Branch 11 had not, till this stage, been recorded in the look-up table, as a terminal branch. A vertical scan of the incidence matrix along branch 11 is therefore needed. The vertical scan yields a -1 at node6. Node 6 is however a terminal node, branch 11 is therefore discarded as a tree branch candidate.

Witn node 4 as a starting node and proceeding as for the two preceeding starting nodes, yields branch 4 as the branch candidate.

Starting node scan stops at node 4 for the example given.

The algoritan thus yields the tree of the network as in Fig.1.5

The formation of the tree indicates the priority ordering of nodes and branches, the lower numbered nodes and branches being given the higher priority during the formation of the network tree.

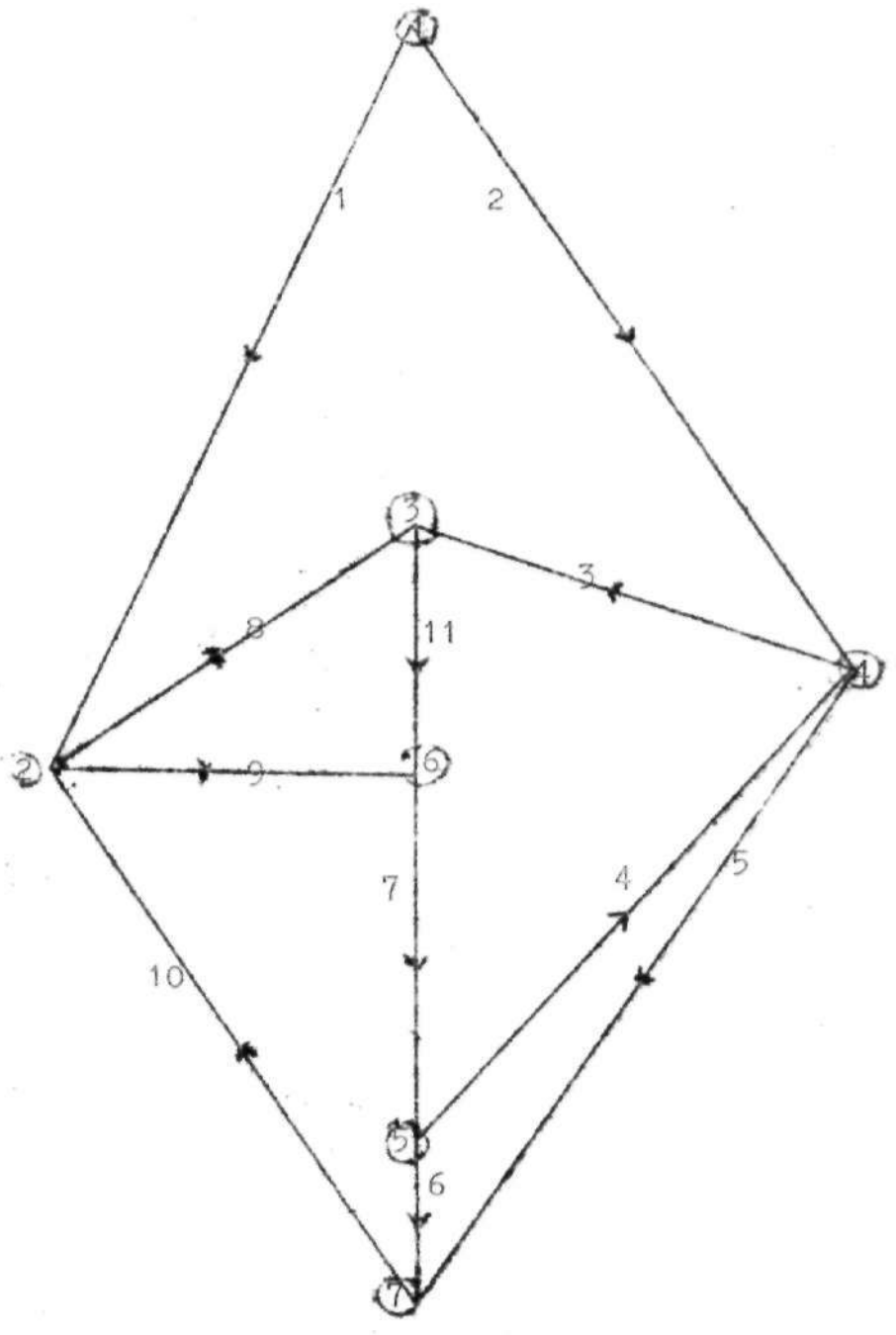


Fig.1.4. Network graph used to illustrate the applications of the 'maxima tree branch per node' algorithm.

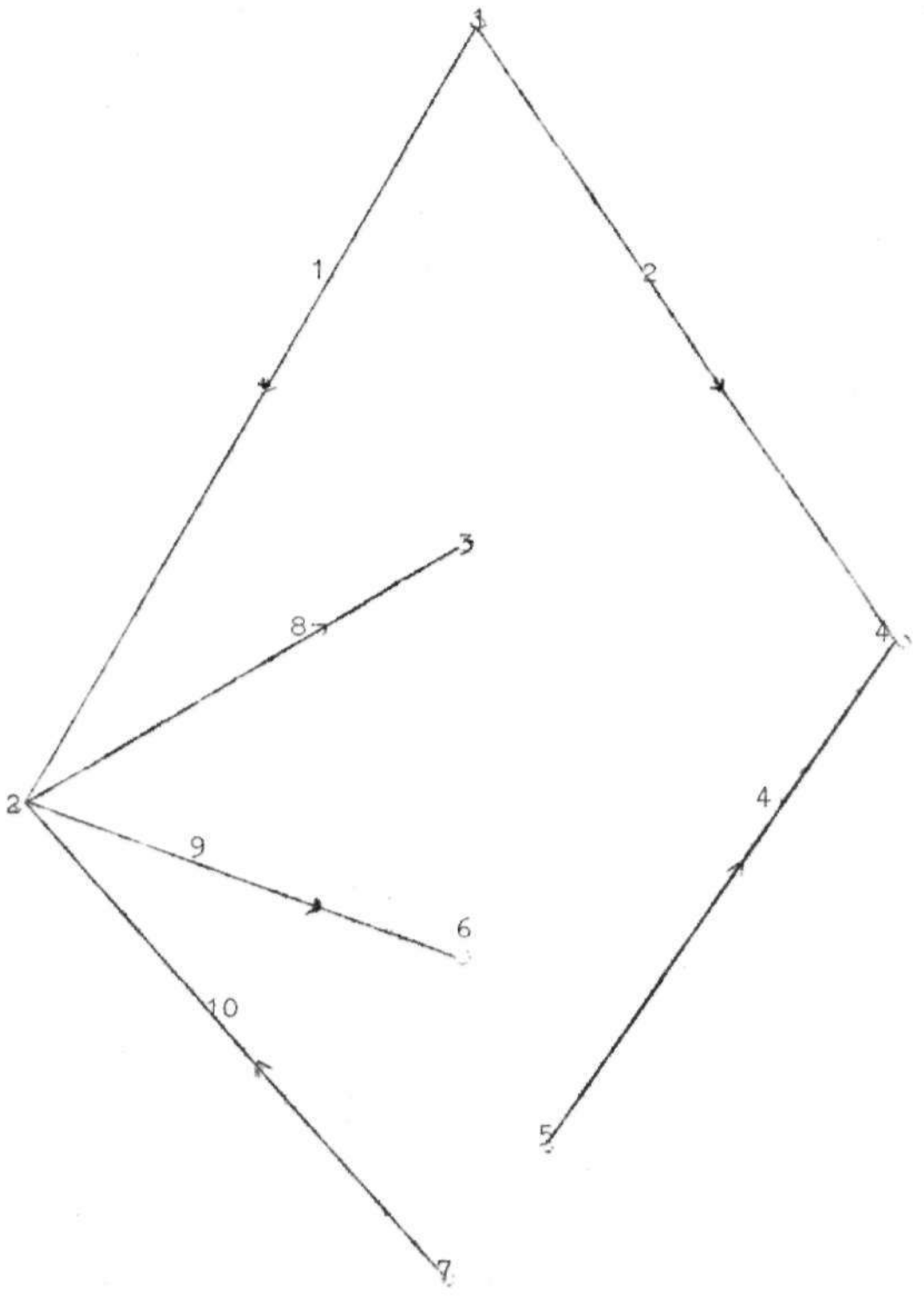


Fig.1.5. Network tree for the graph of Fig.1.4. via the maxima tree-branch per node algorithm

1.4 A SURVEY OF SOME MATRIX METHODS OF NETWORK ANALYSIS

The most common matrix methods for network analysis are :-

(i) cutset analysis

(ii) tieset analysis

(iii) nodal analysis

(iv) mesh analysis

1.4.1 CUTSET MATRIX BASED ANALYSIS[1]

By definition, a cutset is a set of basic cuts, one of which cuts each tree branch, while a cut is understood to mean a closed surface dividing a network into two parts. Consider a network with graph as shown in Fig. 1.6 (the continous lines represent tree branches, the dots represent the links, and the dashes represent the basic cuts).
Kirchoff's current law applied to each cut in the cutset will yield, in general,

$$\underline{D}^t \underline{i} = \underline{0}$$

where $\underline{i}$ is the branch current vector,

$\underline{D}$ is the reduced cutset matrix.

If the link voltages is expressed in terms of tree branch voltages via the standard branch expression, we have:-

$$(\underline{1} - \underline{Y}\ \underline{b}) + \underline{i} = \underline{Y}\ \underline{e}$$

where $\underline{I}, \underline{i}, \underline{Y}, \underline{e}$, and $\underline{E}$ take values as in the standard branch derivation

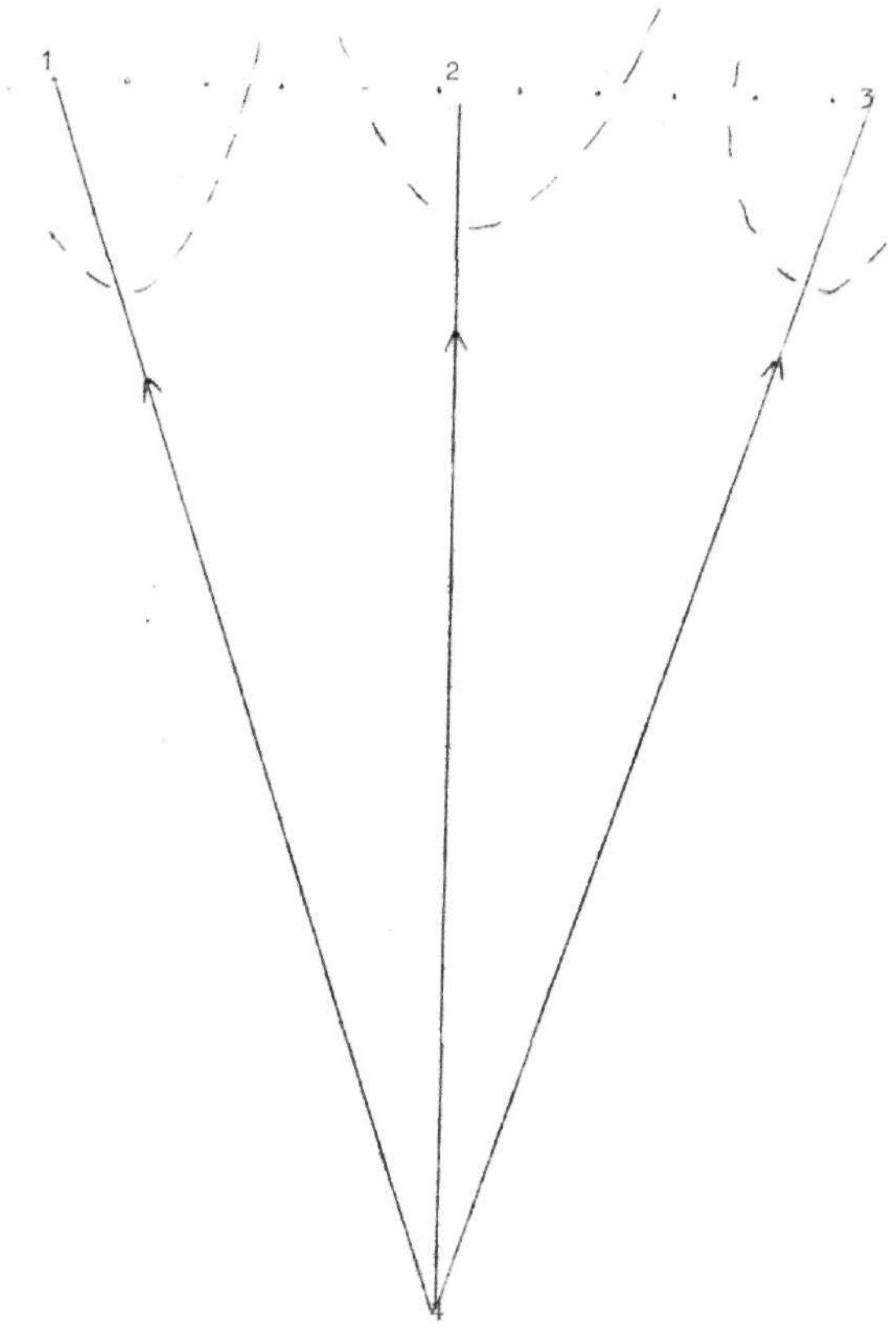


Fig.1.6. Choice of basic cuts for the graph of fig.1.1.1

Multiplying by $\underline{D}^t$ yields

$$\underline{D}^t(\underline{I} - \underline{YE}) + \underline{D}^t\underline{i} = \underline{D}^t\underline{Ye}$$

from which we may deduce

$$\underline{D}^t(\underline{I} - \underline{Ye}) = \underline{D}^t\underline{YDe}_t$$

thus the link voltages $\underline{e} = \underline{D}\ \underline{e}_t$, where $\underline{e}_t$ are the tree branch voltages.

1.4.2 TIESET(LOOP) MATRIX BASED ANALYSIS (1)

Another method used for analysing the abstracted electrical networks is based on the tieset or loop analysis technique.

By definition, a loop is any closed path in a network, while a tieset is a set of basic loops one of which tranverses a link. Fig. 1.7 illustrates the formation of loops. (The continous lines represent the tree branches, the dots represent the links, and the dashes represent the fundamental loops.)

Basically,

number of loops = number of links.

The tieset (loop) analysis technique, expresses branch currents in terms of link currents, viz the tieset analysis technique is the dual of the cutset analysis technique. On the basis of this we may write,

$$\underline{C}\ \underline{i}_l = \underline{i}$$

where $\underline{i}$ is the branch current vector, $\underline{i}_l$ is the link current vector, $\underline{C}$, the reduced loop matrix.

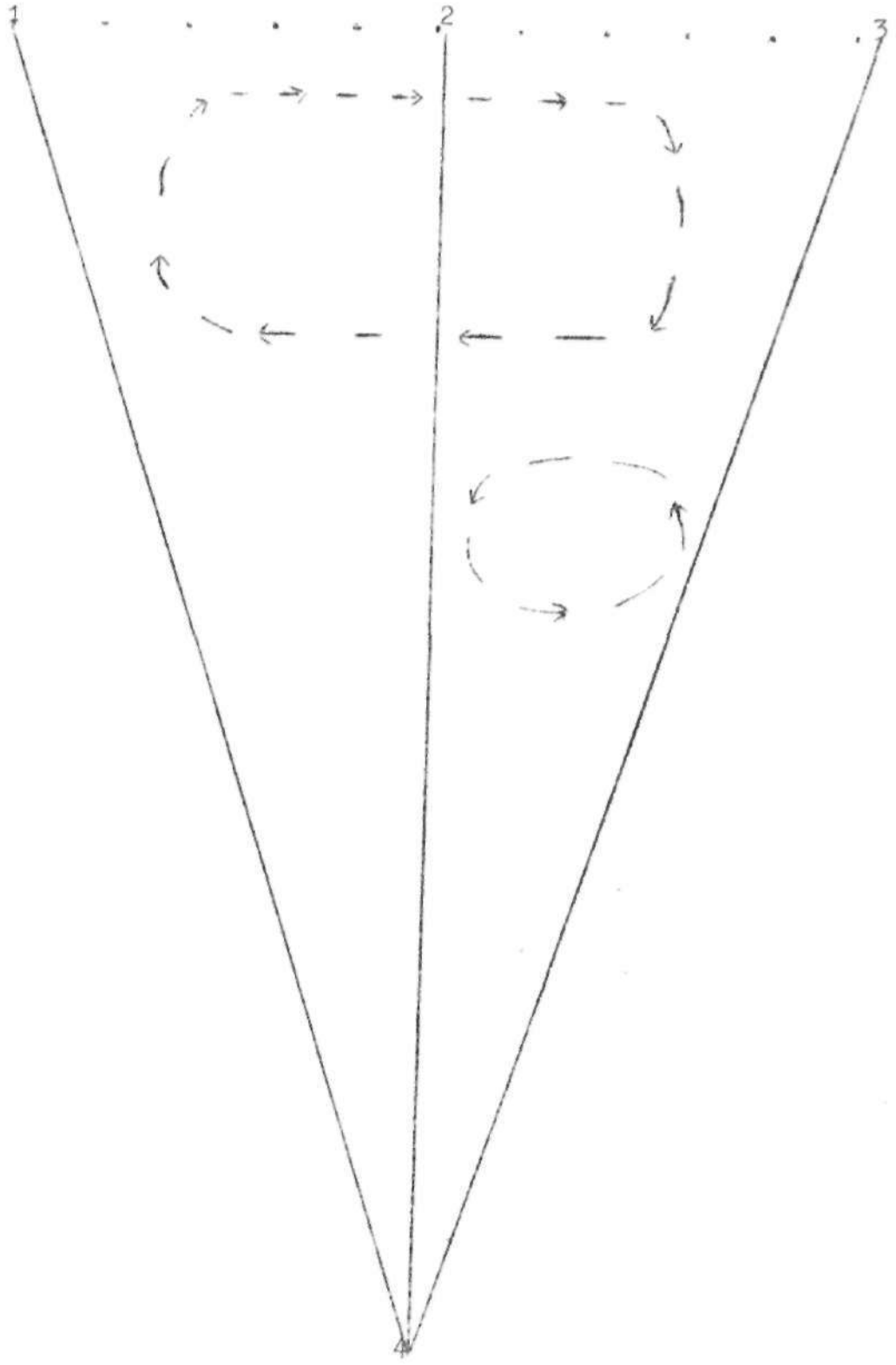


Fig.1.7. Choice of loops(dashed-arrowed) for the graph of Fig.1.1.1

From the standard branch relation,

$$(\underline{E} - \underline{Z}\ \underline{I}) + \underline{e} = \underline{Z}\ \underline{i}$$

Multiplying by $\underline{C}^t$ gives

$$\underline{C}^t(\underline{E} - \underline{Z}\ \underline{I}) + \underline{C}^t\ \underline{e} = \underline{C}^t\ \underline{Z}\underline{i}$$

Applying Kirchoff's voltage law gives

$$\underline{C}^t(\underline{E} - \underline{Z}\ \underline{I}) = \underline{C}^t(\underline{E} - \underline{Z}\ \underline{I})$$

for which we get link currents, $\underline{i}_1 = (\underline{C}^t\underline{Z}\ \underline{C})^{-1}\underline{C}^t(\underline{E}-\underline{Z}\underline{I})$

1.4.3 MESH MATRIX BASED ANALYSIS[1]

Mesh analysis is applicable in the general, only to planar networks.

Fig.1.8 illustrates the formation of meshes.

Basically,

number of mesh loops = number of branches − number of nodes +1

As an example of the special case of the general non-applicability of mesh analysis to non-planar networks, consider the network with graph as in Fig.1.9.

Planes could be chosen such that a node is not traversed by a mesh. (From the general rule, with the eight branches and 5 nodes, 4 meshes would have sufficed).

The example of Fig.1.9 gives 4 chosen meshes indicated by dashed lines.

With node1 not traversed, solution to the node voltages could thus not be obtained from the chosen meshes.

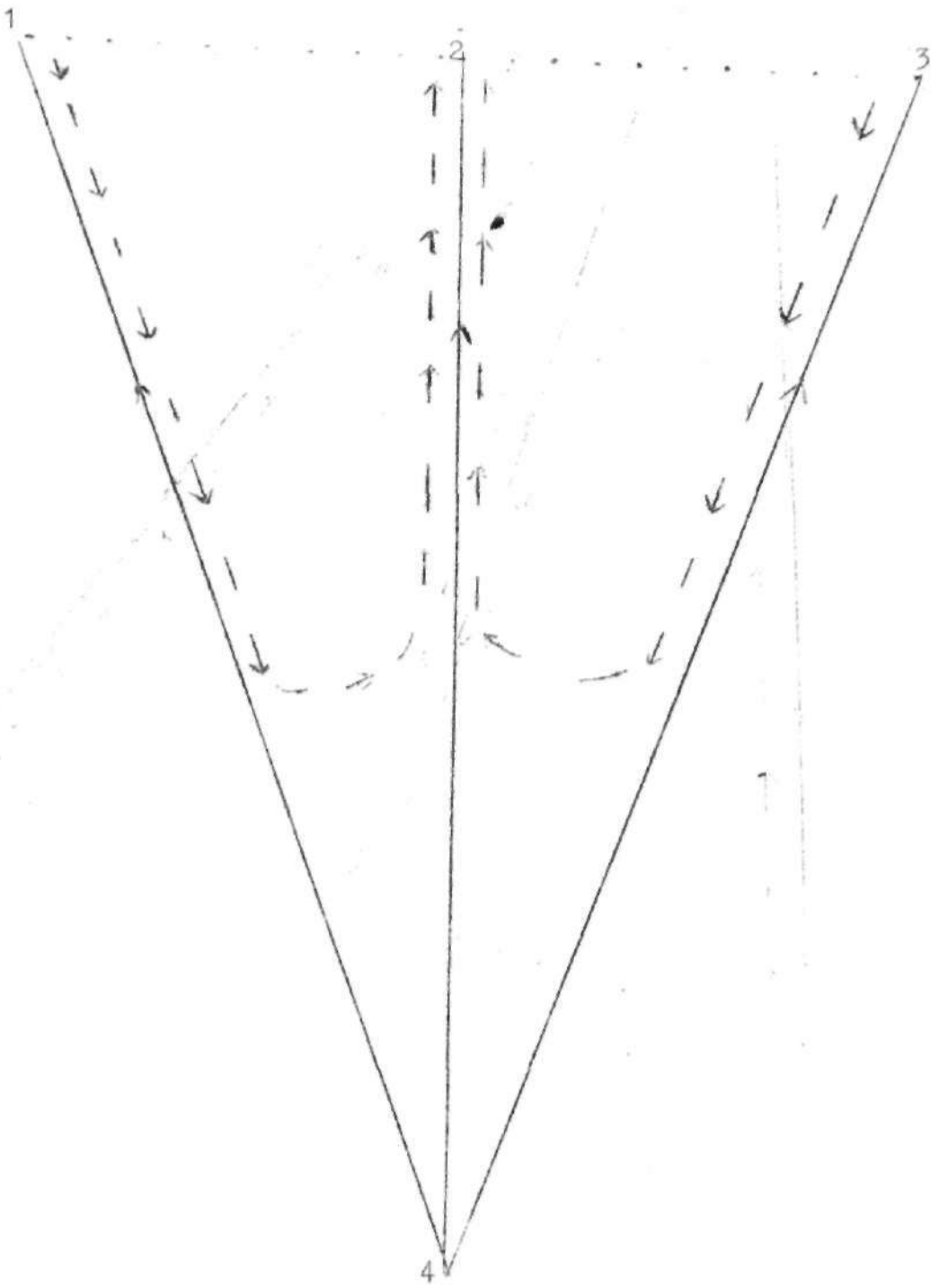


Fig.1.8. Formation of meshes(dashed-arrowed) for network of Fig.1.1.1.

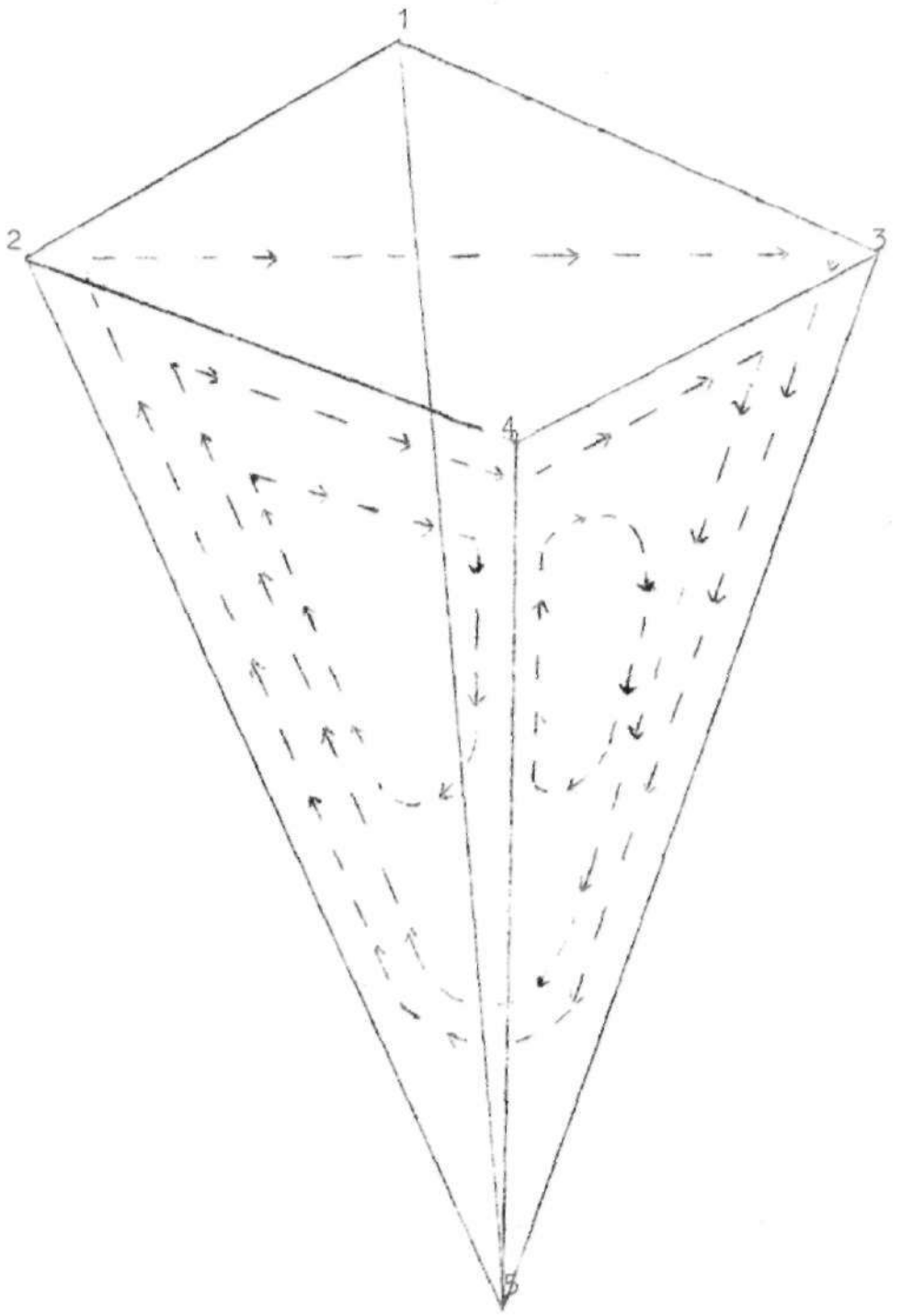


Fig.1.9. Non-planar network to illustrate the non-applicability of mesh matrix method, in the general to non-planar networks-node 1 not traversed. (Chosen meshes are dashed-arrowed.)

Application of Kirchoff's voltage law to the general planar case gives

$$\underline{M}^t = \underline{0}$$

where the superscript, t, indicates the transpose of the reduced mesh matrix $\underline{M}$

If $\underline{i}'$ is taken as the column vector of mesh currents, then

$$\underline{M}\underline{i}' = \underline{i}$$

where $\underline{i}$ is the branch current vector.

Hence,

$$\underline{i}' = (\underline{M}^t\underline{Z}\underline{M})^{-1}\underline{M}^t(\underline{E} - \underline{Z}\underline{I})$$

1.4.4 NODAL MATRIX BASED ANALYSIS TECHNIQUE[1]

This is the method used for the D.C. mode computer aided design program developed in this thesis.

The nodal analysis technique is similar to the cutset analysis technique with the difference that the reduced incidence matrix is replaced by the cutset matrix.

Considering the graph of Fig 1.10 with current directions as indicated, we may write, using Kirchoff's current law for the general case,

$$\underline{A}^t\ \underline{i} = \underline{0}$$

where $\underline{A}$ is the reduced incidence matrix.

If $\underline{e}'$ represents the set of voltages for each node referred to the reference node, then

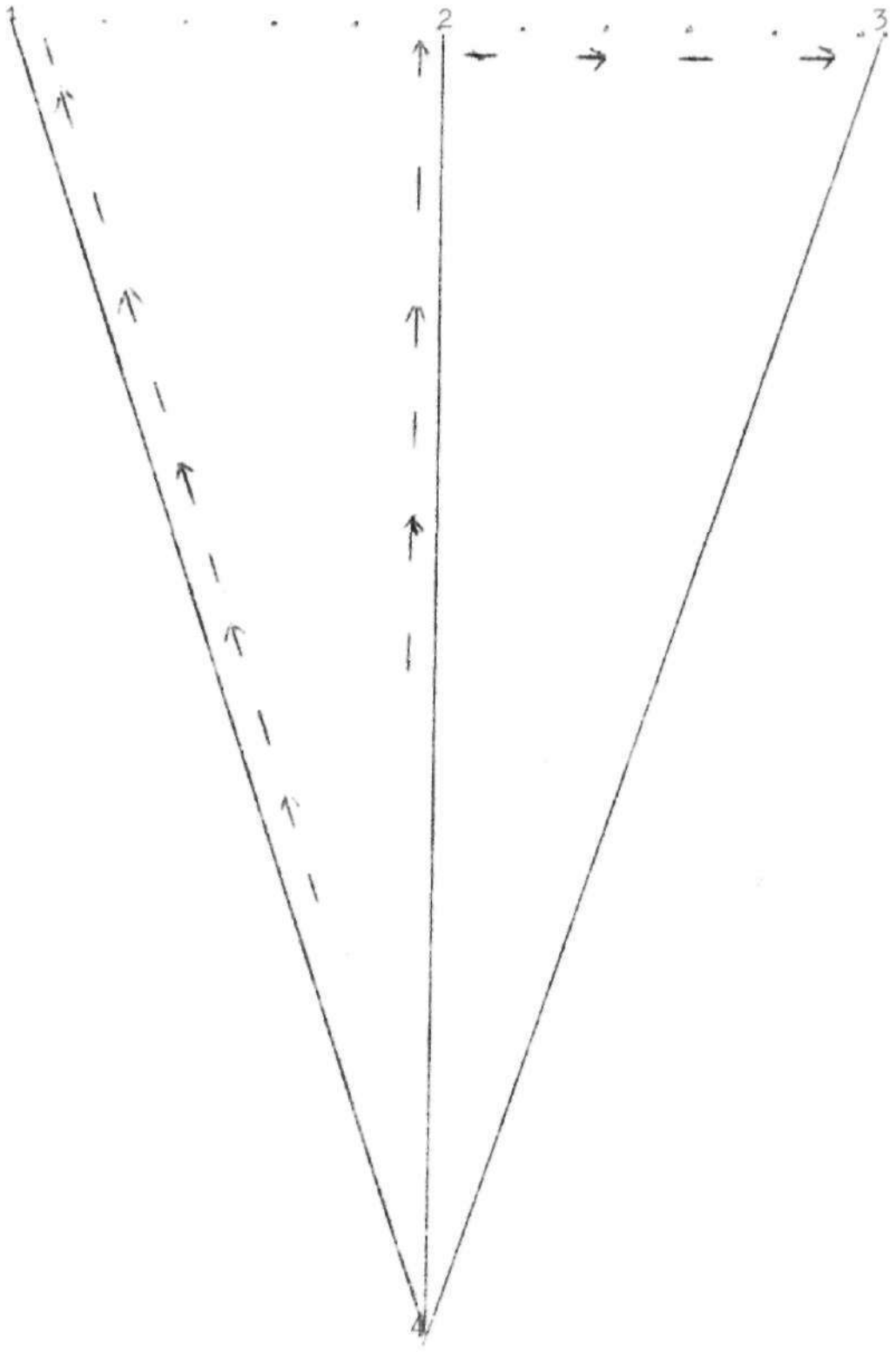


Fig.1.10. Choice of current directions (dashed-arrowed) for nodal analysis withrespect to Fig.1.1.1

$\underline{A}\ \underline{e'} = \underline{e'}$

where $\underline{e}$ is the branch voltage vector.

From the standard branch relation,

$(\underline{E} - \underline{Z}\ \underline{I}) + \underline{e} = \underline{Z}\ \underline{i}$

Multiplying by $\underline{A}^t$ gives

$$\underline{A}^t(\underline{E}-\underline{Z}\ \underline{I}) + \underline{A}^t\underline{Z}\ \underline{i}$$

Now,

$$\underline{A}^t\ \underline{i} = \underline{O}$$

Hence,

$$\underline{A}^t(\underline{I} - \underline{Y}\ \underline{E}) = \underline{A}^t\ \underline{Y}\ \underline{e'}$$

Solving for node voltages yields,

$$\underline{e'} = (\underline{A}^t\ \underline{Y}\ \underline{A})^{-1}(\underline{A}^t(\underline{I} - \underline{Y}\ \underline{E}))$$

1.5 ANALYSIS VIA THE NODAL MATRIX TECHNIQUE - REASONS FOR CHOICE

The following reasons are advanced for the choice of network analysis via the nodal matrix formulation :-

(i) One of the most popular network analysis program , the IBM ECAP program, is formulated on nodal matrix basis and so is the Ohio University "LINCAD" program by Cornetet and Battocletti.

(ii) It would be difficult for the computer to identify meshes in the case of an analysis via mesh matrix method.

(iii) Node potentials are often what is of interest to the circuit analyst.

More-over a simple transformation in the nodal analysis case yields the branch voltages as well.

(iv)The fact that ideal voltage sources cannot directly be incorporated into the program in the analysis via the nodal matrix method is not a major disadvantage, since most physical voltage sources have finite characteristic internal impedances

(v) An analysis via cutset or tieset matrix methods requires the generation of specific trees which may in several cases not be a trivial assumption and except in specific cases, does not serve the teaching requirements which should also include minimisation of time wasted and scarce computer resources.

CHAPTER TWO

THE NEWTON-RAPHSON ITERATIVE APPROACH TO SOLUTION OF NON-LINEAR CIRCUIT PROBLEMS - A SURVEY .

INTRODUCTION

The Newton-Raphson technique used for the circuit analysis program in this thesis, is the generally accepted "modified Newton-Raphson approach"[1,18].

The unmodified Newton-Raphson technique requires, for n simultaneous equations in n unknowns, the evaluation of n^2 partial derivatives, from which we must then solve for increments by solving n simultaneous equations in n unknowns.
The modified Newton-Raphson iterative procedure is derived from the single variable Newton-Raphson technique. Thus, for n non-linear simultaneous equations in n unknowns, the single variable Newton-Raphson is applied n times; once for each variable with the assumption that the other variables are fixed.

Powerful as the Newton-Raphson technique is, it is not guarranteed to converge or it may converge to an undesired solution.

For non-linear circuits to be encountered, using "RESTRICTION METHODS" presented, the method has been found to converge to the desired solution for all standard circuits with un-ambigous solution tested.

Each two terminal non-linear element to be considered may be represented by voltage-current relations of the form

$$\underline{v} = g(\underline{i}) \text{ or } \underline{i} = g(\underline{v})$$

where g is a continous single-valued function of the independent variable. The non-linear device characteristic thus determine the type of mathematical relations required in equation formulation for the computer program.

The bulk circuits to be analysed are assumed to be individually single-valued in both current and voltage and is memoryless. The technique presented is therefore not suitable for analysis of circuits with bistable behaviour or circuits with hysterisis behaviour, since even if the iteration converged, there would be no guarrantee that it is to the desired point in such cases.

2.1 THE MODIFIED NEWTON-RAPHSON PROCEDURE APPLIED TO NODAL CIRCUIT ANALYSIS[18]

Consider the Taylor expansion of an n-variable

function with independent variables

$$\underline{x} = \left[x_1, x_2, x_3, \ldots\ldots\ldots, x_n\right]^t$$

then

$$f(\underline{x} + \Delta\underline{x}) = f(\underline{x}) + g\Delta\underline{x} + \tfrac{1}{2}\underline{x}^t H \Delta\underline{x}$$

where

H = the Hessian of f(x)

g = the gradient vector

If the gradient terms higher than two are neglected, we arrive at the modified Newton-Raphson procedure, viz:-

$$f_1(x + \Delta x) = f_1(x) + g_1 \Delta x$$

$$f_2(x + \Delta x) = f_2(x) + g_2 \Delta x$$

$$\ldots\ldots\ldots\ldots\ldots\ldots$$

$$f_n(x + \Delta x) = f_n(x) + g_n \Delta x$$

The generalised modified Newton-Raphson relations then become

$$\underline{G}\,\Delta\underline{x} = -\underline{f}.$$

In circuit analysis $\underline{x}$ corresponds to node voltages $\underline{e'}$.

Making use of the standard branch relation gives :-

$$\underline{f}(\underline{e}) = \underline{A}^t \underline{Y}(\underline{e'})(\underline{A}\,\underline{e'} + \underline{E}) - \underline{A}^t\underline{I} \quad \ldots\ldots 2.1$$

Now,

$$\underline{j} = \underline{i} + \underline{I} = \underline{Y}(\underline{e} + \underline{E}) = \underline{Y}(\underline{A}\,\underline{e'} + \underline{E})$$

$$\underline{v} = \underline{e} + \underline{E} = \underline{A}\,\underline{e'} + \underline{E}$$

Substituting into 2.1 and differentiating gives

$$\underline{G} = \underline{A}^t \frac{d}{dv}(\underline{Y}\,\underline{v})\,\underline{A} = \underline{A}^t\underline{Y}\,\underline{A}$$

where $\underline{Y}$ is the linearised admittance matrix.

2.2 CONVERGENCE PROBLEMS OF THE NEWTON-RAPHSON ITERATIVE PROCEDURE AND METHODS USED IN THE THESIS TO OVERCOME THEM (18)

For ease of presentation, assume a single non-linear component, a diode, with voltage-current characteristics and load line as shown in Fig.2.1. (An n-component characteristic would have a representation in n+1 dimensional Euclidean space).
Starting at point (i_o, v_o) and proceeding in the direction indicated gives an "oscillating convergence"- a typical case with diode non-linearities.
With the same characteristic, but starting at a point close to the true solution point as indicated in Fig. 2.2 results in an oscillating divergence such that the true operating point of the diode is never found. (18) The starting point of an iteration as well as its direction, is crucial at obtaining convergence.

A case in which an upper and lower bound is placed on the possible iterative voltage values is shown in Fig. 2.3. The characteristic as well as the starting point of Fig. 2.2. is retained.
It may be visualised that placing bounds on the voltage values results in "oscillating convergence"(18).
INSIGHT : To enhance convergence, place upper and lower bounds on the diode junction voltages.

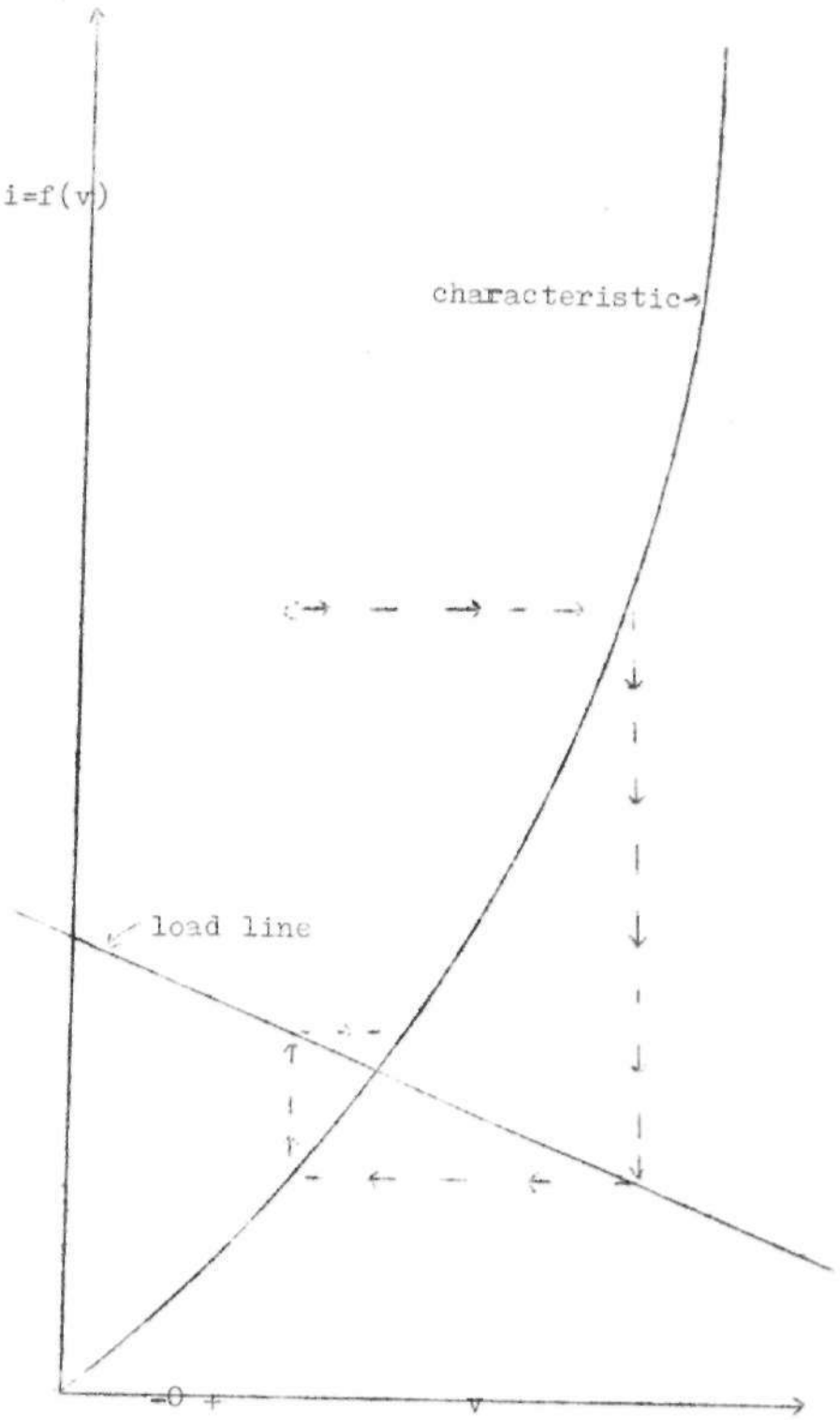


Fig.2.1. Illustration of "oscillatory convergence" with direction(dashed-arrowed) of iteration, though starting point close to true solution point.

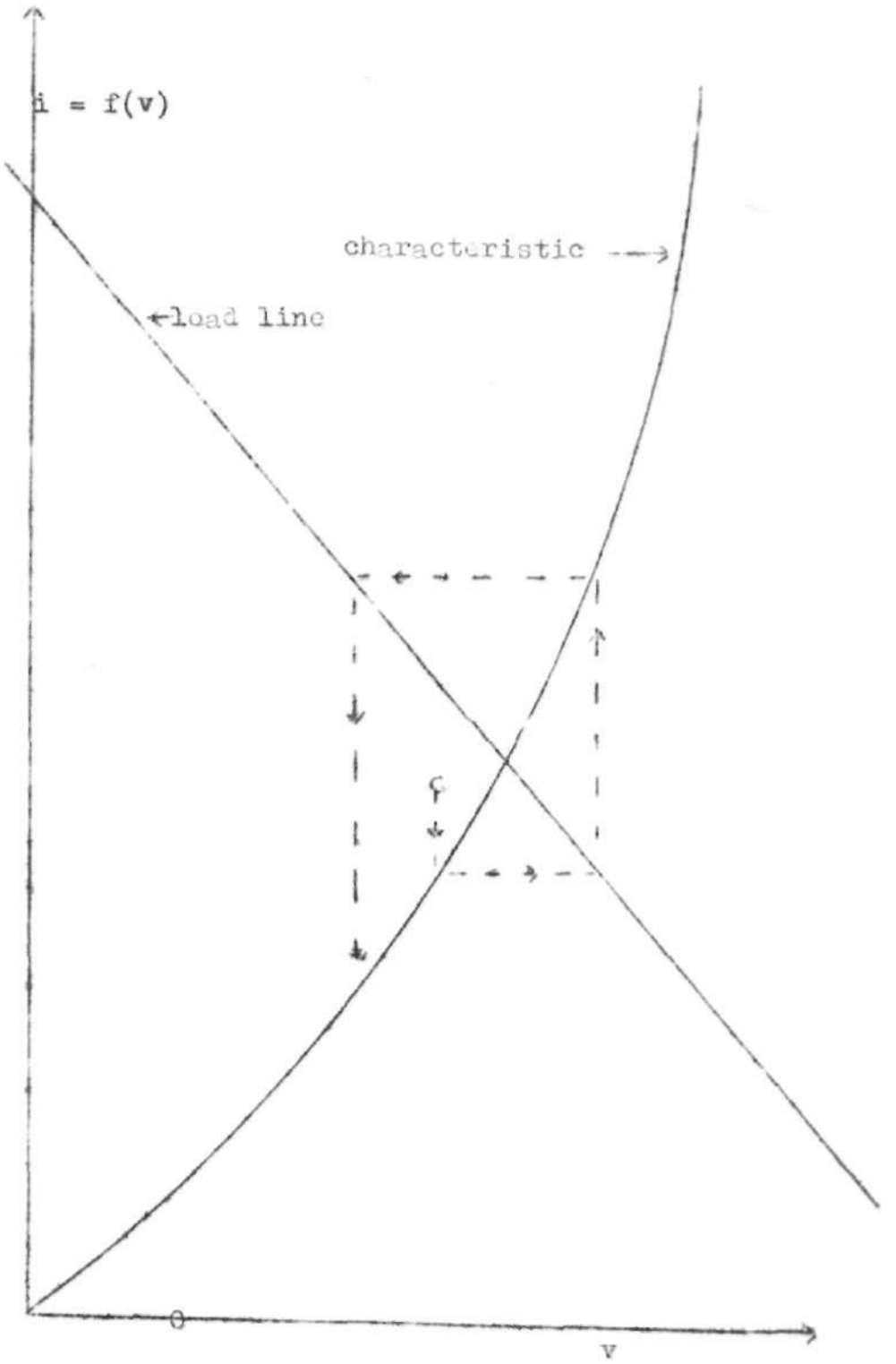


Fig.2.2 Illustration of "oscillatory divergence" with direction (dashed-arrowed) of iteration, though starting point close to true solution point.

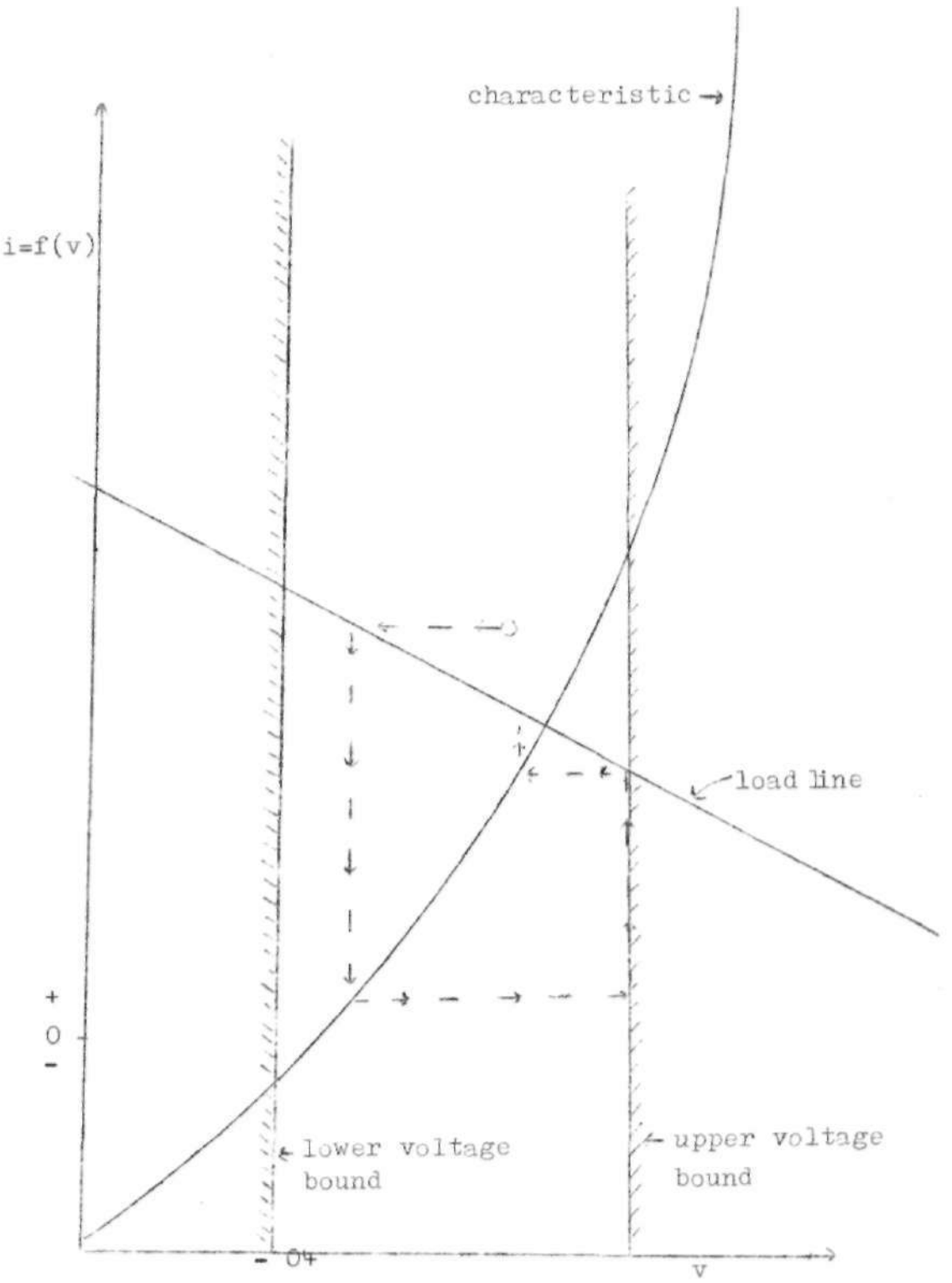


Fig.2.3. The use of bounding to ensure convergence in an otherwise divergent iteration -an upper bound for forward biased junctions and a lower bound for reverse biased junctions.

A close look at the derivation of Newton-Raphson iteration via Taylor's series, viz:-

$$f(e_{i+1}) = f(e_i) + \left(f'(e_i) \; \partial f / \partial e \right) \Big|_{e=e_i} (e_{i+1}-e_i)$$

suggest we use the mean value theorem to discover the bounds that would result in convergence of the Newton-Raphson iteration. From the statement of the mean value theorem, if a function, f(e) and its derivative, f'(e) are both continous in the interval $a \leq e \leq b$, there exists at least one $a < \psi < b$ such that,

$$f'(\psi) = \frac{f(b) - f(a)}{b-a}$$

viz, $f(b)-f(a) = f'(\psi)(b-a)$.

A graphical interpretation withrespect to diode characteristic is presented in Fig. 2.4.

If we apply the mean value theorem to our iteration guess,

$$e_{i+1} = f(e_i).$$

Letting $a = e_{i-1}$, $b = e_i$

then

$$f(e_i) - f(e_{i-1}) = f'(\psi)(e_i-e_{i-1})$$

where ψ lies between e_i and e_{i-1}.

Now,

$$f(e_i) = e_{i+1}, \quad f(e_{i-1}) = e_i.$$

Hence we may write

$$e_{i+1} - e_i = e_i - e_{i-1}$$

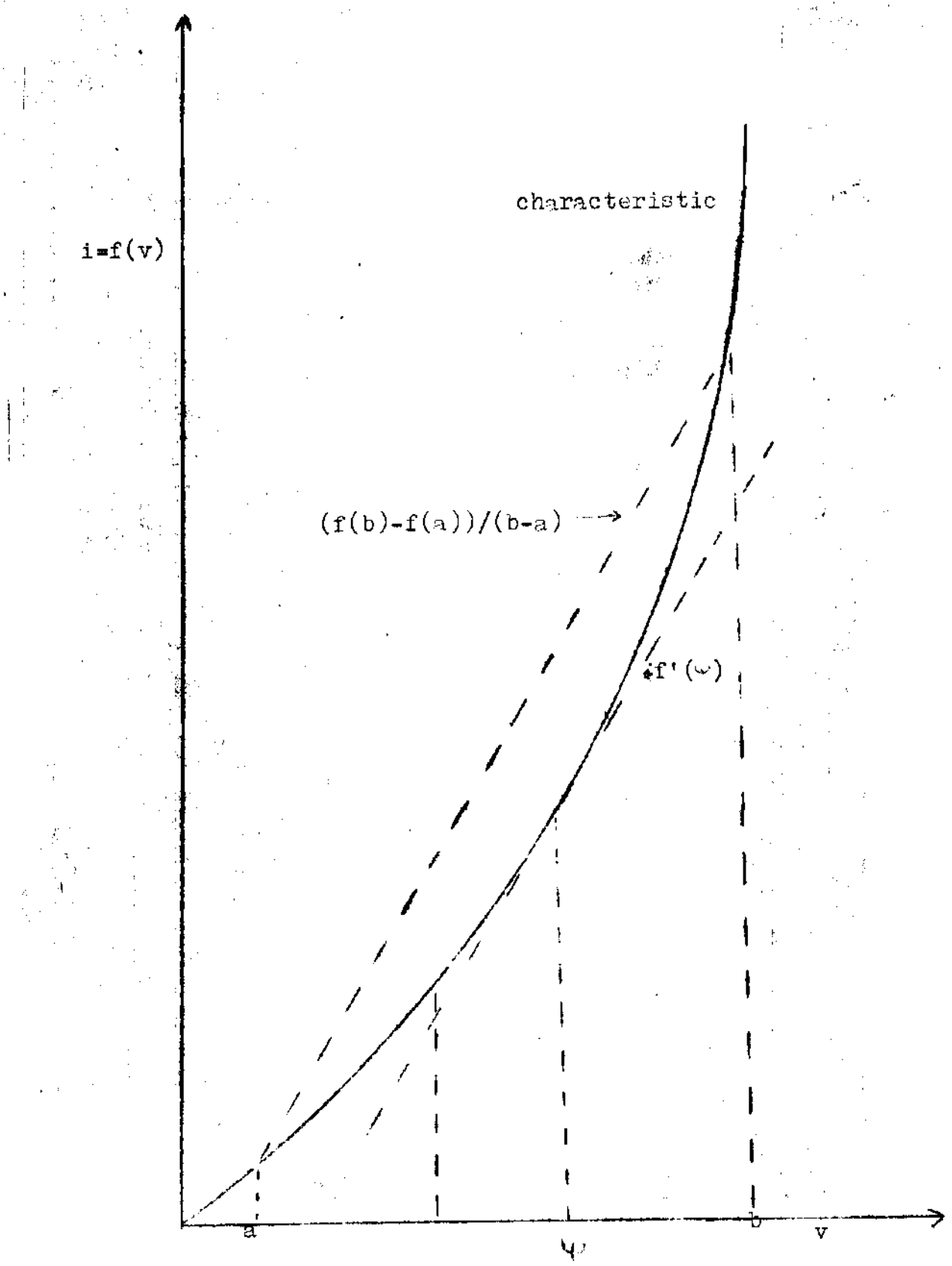


Fig.2.4 Application of mean value theorem to diode characteristic in the derivation of convergence criterion.

Applying the mean value theorem to each iterative sequence gives

$$|e_2 - e_1| = f'(\Psi_1) \; |(e_1 - e_0)|$$

$$|e_3 - e_2| = f'(\Psi_2) \; |(e_2 - e_1)|$$

.....

$$|e_{i+1} - e_i| = f'(\Psi_i) \; |(e_i - e_{i-1})|$$

Now we had placed a bound on all of e_i. Assuming also a bounded actual root $\underline{e}_b$, and a bounded $f'(e)$, then we may write for any i and some upper bound M,

$$f'(\Psi_i) \leq M$$

Hence

$$|e_2 - e_1| \leq M \leq |(e_1 - e_o)|$$

$$|e_3 - e_2| \leq M \leq |(e_2 - e_1)|$$

.....

$$|e_{i+1} - e_i| \leq M \leq |(e_i - e_{i-1})|$$

Substituting the first relation into the second and the result into the next e.t.c. yields

$$|e_4 - e_3| \leq M \; |(e_3 - e_2)$$

$$\leq M.M^2 |(e_1 - e_0)|$$

.....

$$|e_{i+1} - e_i| \leq M^i \; |(e_i - e_o)|$$

Now for convergence

$$\lim_{x \to \infty} x_{i+1} - x_i = 0$$

The above condition may be forced to exist by making $M < 1$ or $M^i \to 0$ as $i \to \infty$

Hence the iteration must converge if $M < 1$.

Thus if $|f'(e)| < M < 1$ for all x, we have a sufficient though not necessary condition for the iteration convergence.

If f'(e) is near zero in the entire region, the iteration converges quickly, while if it is near 1 in magnitude, it converges slowly.

If we now go back to the Newton-Raphson iteration formula $e_{i+1} = e_i - (f(e)/\partial f/\partial e)\Big|_{e=e_i}$ we immediately notice that the second term on the R.H.S. represent the incremental elemental conductance in circuit analysis.

From the above derivation, the iteration converges if

$$|f(e')| < 1$$

and converges very fast if

$$|f'(e)| \to 0$$

viz if the elemental conductance approaches infinity.

INSIGHT 2 : For fast convergence of the iteration, scale or damp the iteration such that the iterative conductance is proportionately very large for each elemental value. The iteration damping factor should be less than unity. The easiest way to achieve this is to scale the impedance value by say 10^{-3}, thus resistance values are best expressed in kilo-ohmns.

In order to retain actual circuit voltages, all circuit currents including voltage controlled current sources (VCCS) must be divided by impedance scale factor.

Consider the diode characteristics with load line as shown in Fig. 2.5.

Consider the iterative point(i_1, v_1) on the characteristic such that v_1 lies positive of v_o, the load voltage.

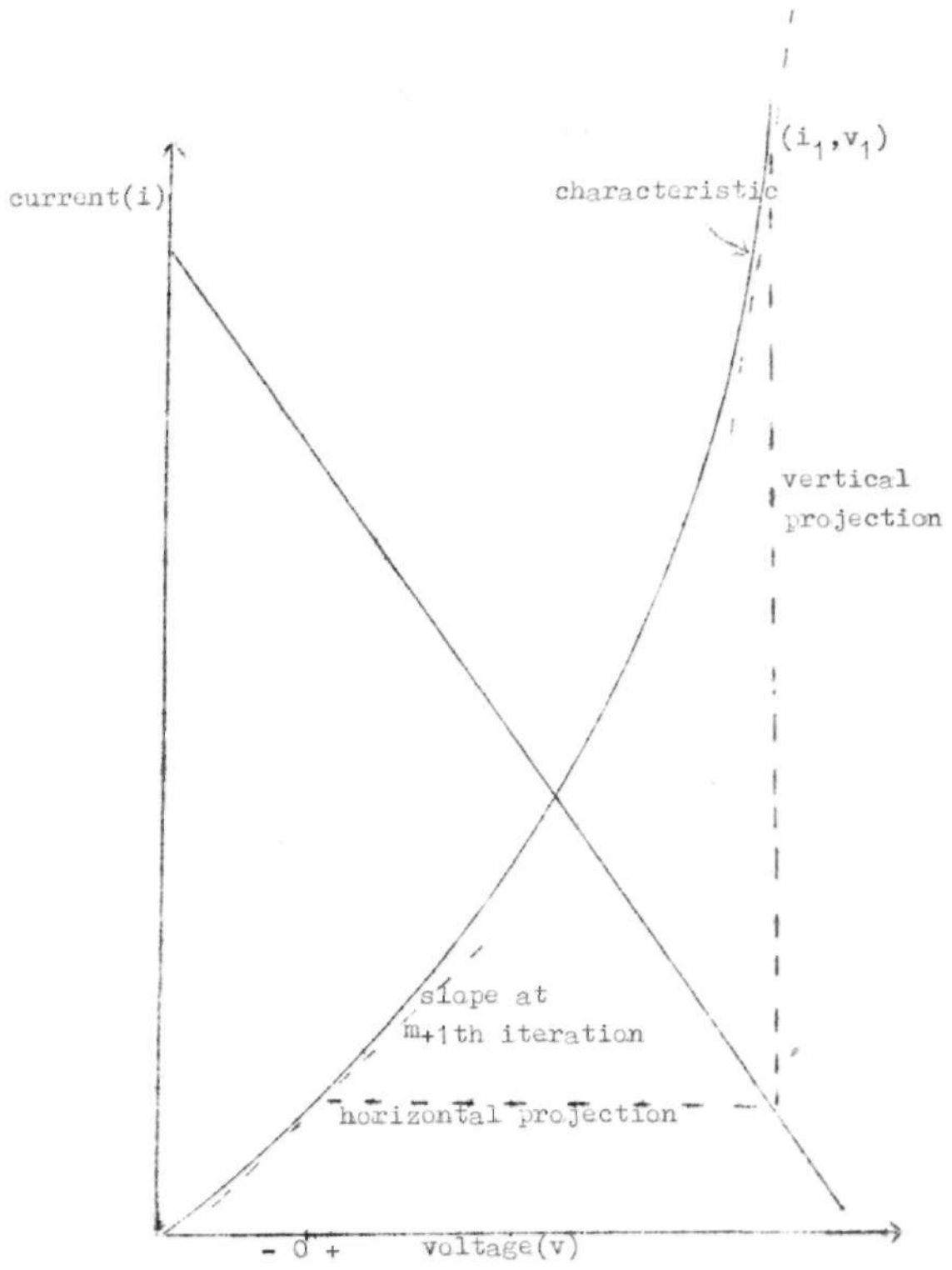


Fig.2.5 Choice of iterative voltage values to enhance convergence

It is easily visualised that if a new updated iterative voltage is calculated to be greater than previous value, we take a vertical projection from the iterative solution point and let the new iterative voltage equal the old one, the iteration is forced to converge.

INSIGHT3: For changes in voltages that are positive, use the old voltage value, the new conductance value being the slope of the i-v characteristic at the old voltage value.

CHAPTER THREE

ITERATIVE DEVICE MODELLING FOR SOME NON-LINEAR RESISTIVE DEVICES

INTRODUCTION

The iterative models for the non-linear devices are meant for the modified Newton-Raphson iterative procedure. Iterative models are presented for practical diodes, transistors, and MOSFETs. It is shown that any three terminal device with dependent controlled source, may be modelled in a way similar to the Ebers-Moll transistor model.

Consider a non-linear device, with current-voltage relation given by $i=f(v)$, where i is single valued in v. A Taylor's series expansion as used for the Newton-Raphson gives a linearised iterative form [2,1]:-

$$i_{m+1} = i_m + \partial f/\partial v\big|_{v=v_m}(v_{m+1} - v_m)$$

Now $i = f(v)$

$$i_{m+1} = i_m + (\partial i/\partial v)_m(v_{m+1} - v_m)$$

But $(\partial i/\partial v)_m$ represents the conductance at the mth iteration. Hence we may write :-

$$i_{m+1} = i_m + G_m(v_{m+1} - v_m) \quad \ldots\ldots\ldots \quad 3.1$$

Expression 3.1 gives an electrical analog as presented in Fig. 3.1.

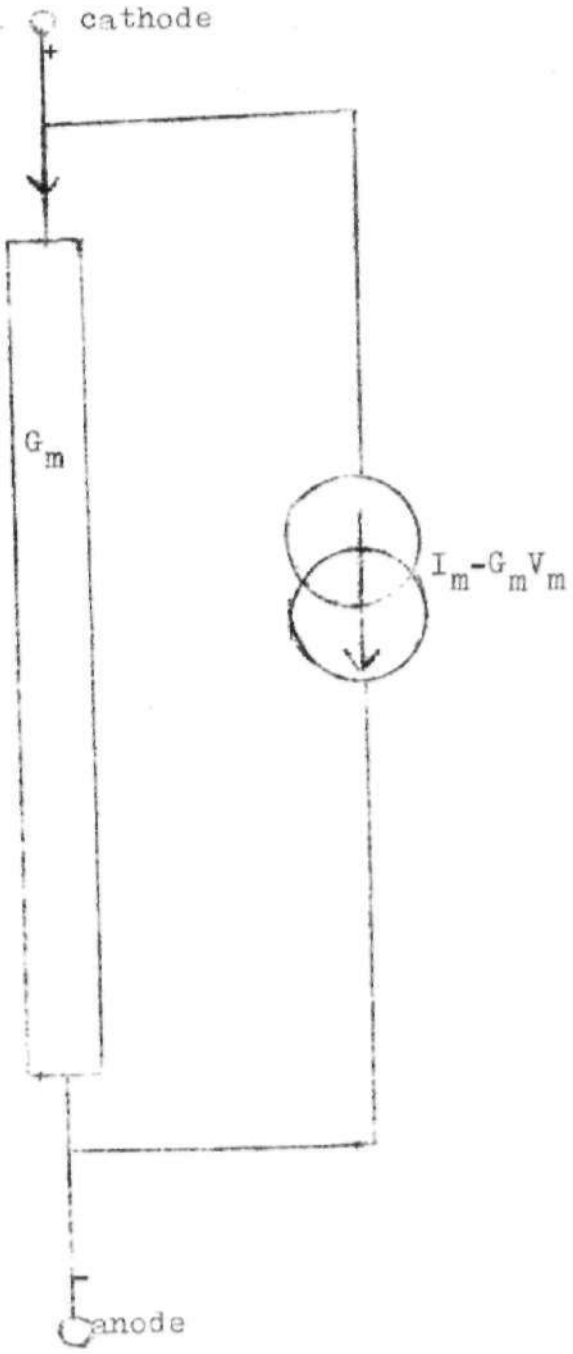


Fig.3.1. Non-linear device characterisation.
(reverse biased).

3.1 THE ITERATIVE DIODE MODEL[2]

The diode current-voltage characteristics may be expressed thus :-

$$i = I_{es} \exp(qv/kt)$$

where q = electron charge = 1.602×10^{-19} coulomb

k = Boltzman constant = $1.38 \times 10^{-23} J/^{o}K$

Differentiating gives ;

$$di/dv = I_{es} e^{qv/kt}(q/kt)$$

Letting $q/kt = \lambda$

gives $di/dv = \lambda I_{es} \exp(\lambda v)$ 3.2

Substituting 3.2 into 3.1 gives

$$i_{m+1} = i_m + \lambda I_{es} \exp(\lambda v)(v_{m+1} - v_m)$$ 3.3

The iterative diode conductance G_m is given by $G_m = \lambda I_{es} \exp(\lambda v_m)$

Expression 3.3 gives the iterative equivalent of the diode as in Fig.3.2.

3.2 THE EBERS-MOLL INTRINSIC BIPOLAR TRANSISTOR MODEL[2,4,12]

The Ebers-Moll intrinsic transistor model is chosen because of its suitability to computer applications. A form of the model that is applicable to arbitrary junction voltages is given in Fig.3.3.

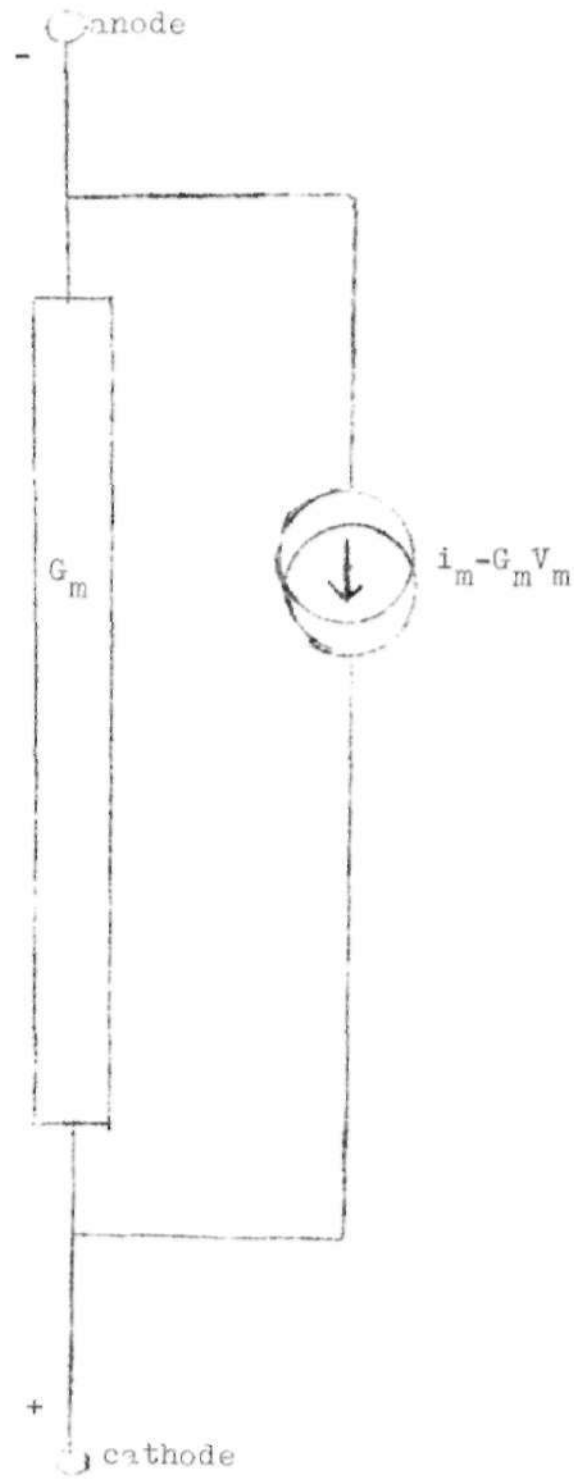


Fig.3.2. The iterative diode model.

For an n.p.n transistor we may write

$$I_e = -I_{es}(\exp(-qv_{eb}/kt)-1)+\alpha_r I_{cs}(\exp(-qv_{cb}/kt)-1)$$
$$= i\ I_{ef} + I_{er}$$
$$I_c = \alpha_f I_{es}(\exp(-qv_{eb}/kt)-1)-I_{cs}(\exp(-qv_{cb}/kt)-1)$$

where

I_{es} = static saturation current with B-C junction shorted.

If we now replace each diode with its iterative equivalent model given in Fig.3.4; for an n.p.n transistor we have

$$I_r = I_{cs}(\exp(-qv_{eb}/kt)-1)$$

$$I_f = I_{es}(\exp(-qv_{cb}/kt)-1)$$

If we define $\lambda = q/kt$, then from Fig.3.4

$G_{1m} = \lambda(I_{fm} - I_{es})$; $G_{2m} = \lambda(I_{rm} - I_{cs})$

$i_{1m} = v_{1m}G_{1m} - I_{fm}$; $i_{2m} = v_{2m}G_{2m} - I_{rm}$

A similar derivation holds for p.n.p bipolar transistors with proper polarity considerations.

3.3 THE ITERATIVE DEPLETION MODE TRANSISTOR MODEL[3,4,1]

The current-voltage relationship for a MOSFET transistor may be expressed as

$$i_d = K(f(v_g,v_s)-f(v_g,v_d))$$

where

v_g = gate voltage, v_s = source voltage

v_d = drain voltage

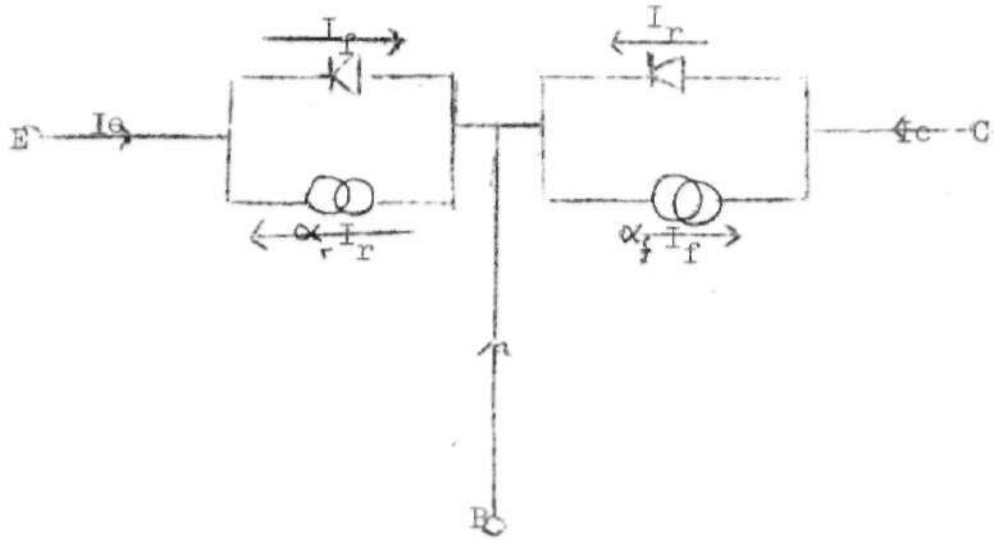


Fig.3.3. Static form of the Ebers-Moll n.p.n transistor model.

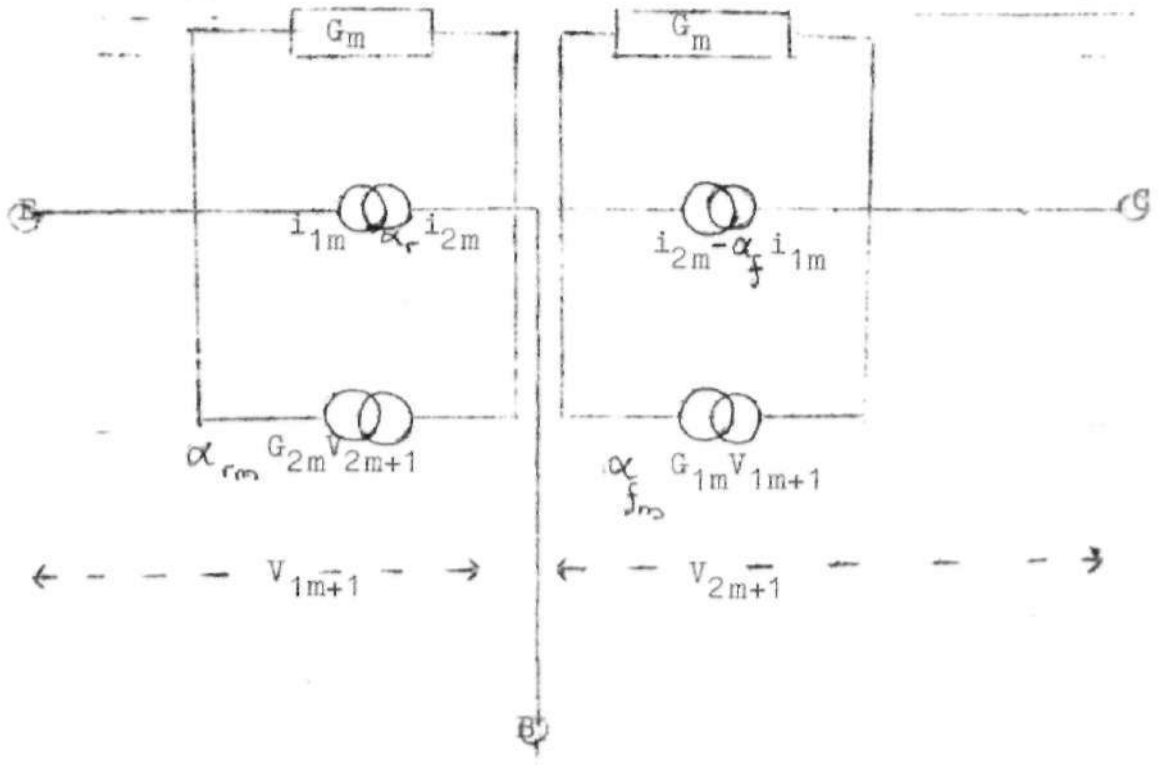


Fig3.4. The iterative model for the n.p.n. transistor.

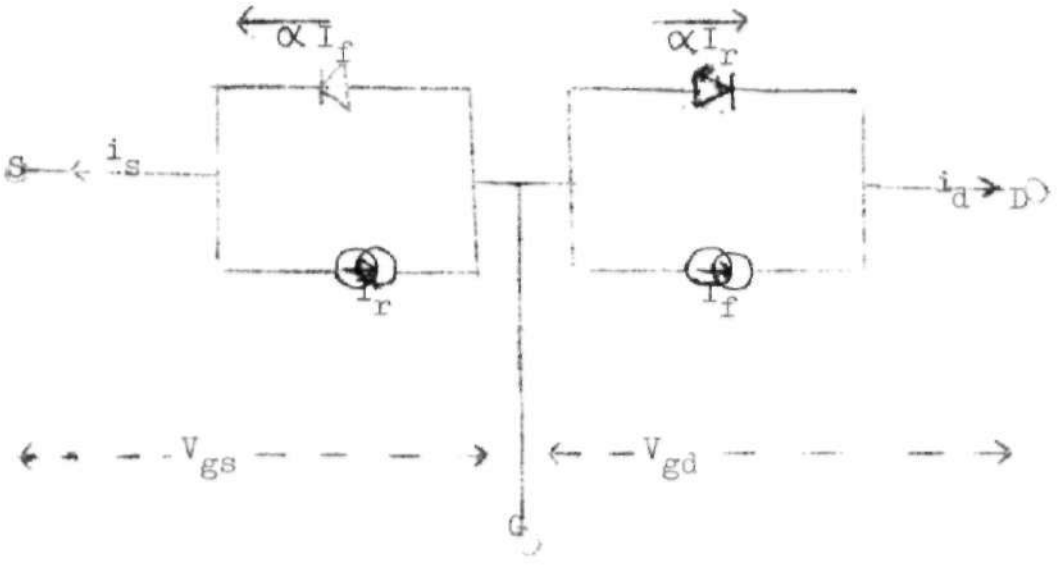


Fig.3.5. An equivalent Ebers-Moll model for the MOSFET.

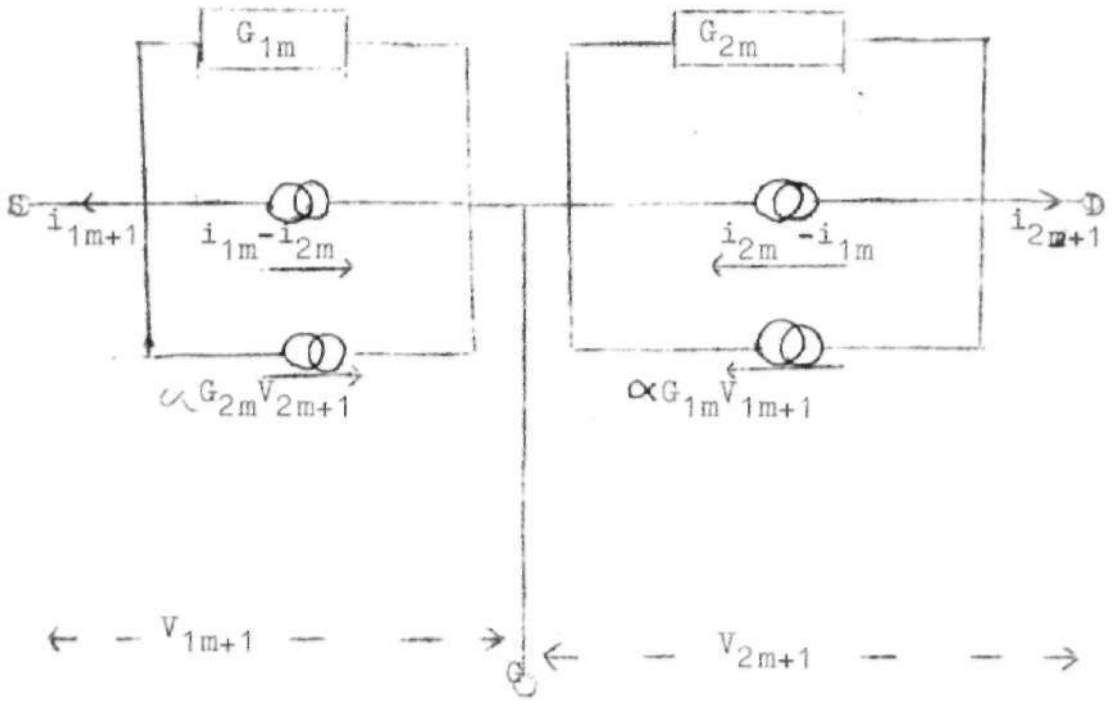


Fig.3.6. The iterative equivalent for the enhancement MOSFET

This is explicitly expressed by the equation

$$i_d = K(v_g - v_p - v_s)^2 - (v_g - v_p - v_d)^2$$

where

v_p = the pinch off voltage

K = constant

For an enhancement MOSFET , if v_{ds} is comparable to v_{gs}, (2,1)

$$I_f = K(v_{gs}v_{ds} - v_{ds}^2/2) \text{ for } v_{ds} \leq v_{gs},\ v_{gs} \geq 0$$

If $v_{ds} > v_{gs}$, the static drain characteristics become

$$I_r = \pm K(v_{gs}^2/2);\ v_{ds} > v_{gs},\ v_{gs} \geq 0$$

For depletion mode MOSFETs the above relationships are adapted by replacing v_s by $v_{gs} - v_p$.

Hence

$$I_f = \pm K((v_{gs} - v_p)v_{ds} - v_{ds}^2/2)$$

for $v_{ds} \leq v_{gs} - v_p$,

$$v_{gs} - v_p \geq 0$$

$$I_r = \pm K((v_{gs} - v_p)/2)^2$$

for $v_{ds} > v_{gs} - v_p \geq 0$

where the + sign refers to n-channel devices and the - sign refers to p-channel devices.

This gives a MOSFET model similar to bipolar model where the I_f and I_r are "diode" currents analogous to the bipolar case, the polarities being dependent on the type of device. Thus for iterative analysis, the bipolar diodes in the bipolar model, are replaced by MOSFET "diodes" in an equivalent model. This is illustrated in Fig3.5. Fig.3.6 gives the iterative model.

CHAPTER FOUR

CONSTRUCTION AND FORMULATION OF THE GENERAL D.C. MODE ELECTRONIC NETWORK ANALYSIS PROGRAM

4.1 INTRODUCTION

The Computer Aided Design of Electronic Networks presented in this thesis is a general purpose D.C. mode resistive network analysis program that allows a wide range of circuits to be analysed - whether linear or non-linear.

Tools incorporated into the program allow ease of simulated breadboarding. Individual component values may be modified, or even removed with the program running. Circuit performance could thus be studied, as is done with prototype breadboarding in the development laboratory. The program is written to be general and with future expansion in mind. Dedicated methods, such as sparse matrix techniques are purposely avoided. Complete program overlays are used for all but two subroutines, thus making for a very efficient program withrespect to memory storage requirements.

Efforts have been geared towards ease of circuit data input and error checking.

The interactive nature of the program allows

the user to use the program without having to memorise procedures or format types as is the case with most elementary programs.

4.2 ATTRIBUTES OF THE PROGRAM

(i) The program is written to be on-line interactive. This means that the program communicates with the user directly on inputs, commands and outputs.

(ii) Inputs are constructed to be format free. This allows rapid turn-arround of large data feeds, since the user need not worry about spaces or length of individual inputs.

(iii) Program construct is such that program data may be saved, (if the user so wishes), with the program running . Data so saved are as permanent files which may then be recalled at a later period, saving the user the need to type in such data when a program based on such data is to be analysed.

(iv) User commands and responses to which the program responds include, "replace","remove","yes",and"no", and also"change"."yes or "no" are desired user responses to interactive program requests such as

(a) "Is operating temperature room temperature?"

(b)"New analysis ?"

(c) "Want to change value of or remove a branch ?"

"Change" is an interactive user command to the program to replace a component impedance value and source voltage with new values typed at the

computer prompt. "Remove" command by the user causes the program to null theeffect on the circuit of the component so indicated. In this way the effect of a single component variation on circuit operation may be studied.

(v) Circuit data, as used by the program are automatically reproduced on the screen or as hard copy before output of results of analysis. This allows rapid turn-around of circuit data input error detection. It may be appreciated that data input errors which are fed in with the programming running, may also be corrected with the program running by the use of the "change" or " remove" commands inbuilt into the program.

(vi) Isolated node checking facility is incorporated into the program which outputs the warning"NODE,X,IS DANGLING FROM CIRCUIT", where X is the node number; ondetection of an isolated node. This inbuilt procedure also enhances data input error checking. The warning could be ignored if such isolated nodes are introduced deliberately by the user, since the program will execute if the circuit has unabigous solution.

(vii)Alphanumerics, which are reproduced on the assigned output device as part of the circuit are used to identify component types; ie, "RES" to indicate that the component type whose identity are to be input is a resistor. Similarly, "DIO" is used for diodes, and "TE" for transistors. "NPN" is used to indicate that the type of bipolar transistor is the npn type.

the user to use the program without having to memorise procedures or format types as is the case with most elementary programs.

4.2.1 THE MAIN PROGRAM

The basic flow-chart for the main program is shown in Fig.4.1. The listing of the D.C. mode computer aided design of electronic networks program is given in Appendix IV. A description of the main program functions are also given in Appendix IV.

Contained in the main program construct are generation of the network incidence matrix, iterative primitive admittance matrix for diodes, bipolar transistors and FETs, as well as the associated nodal current and voltage vectors.

4.2.2 THE DIODE MODELLING ROUTINE

Suppose the element of the jth branch is a diode. From the model of Fig.3.2, the iterative procedure flowchart is presented in Fig.4.2.

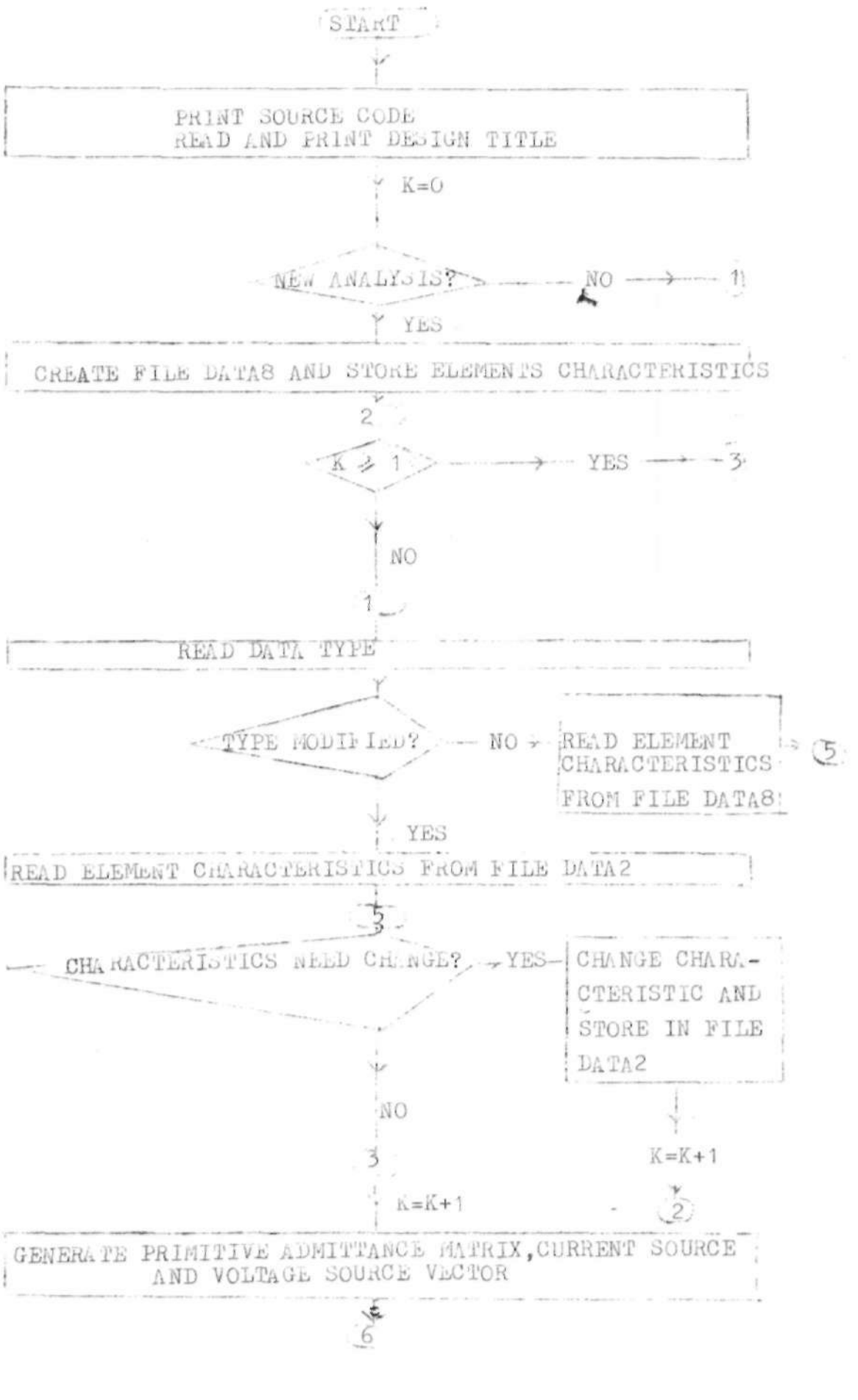
4.2.3 THE TRANSISTOR MODELLING ROUTINE

Assume the B-E junction represents the jth branch, and the C-B junction the (j+1)th branch. The iterative procedure flowchart for the B-E junction is given in Fig.4.3.where v6(j) is the old B-E voltage, vd is the current B-E junction voltage.

A similar procedure holds for the B-C junction.

A similar line of reasoning holds for FETs.

PROGRAM FLOW-CHART FOR THE GENERAL D.C. MODE ELECTRONIC CIRCUIT ANALYSIS PROGRAM



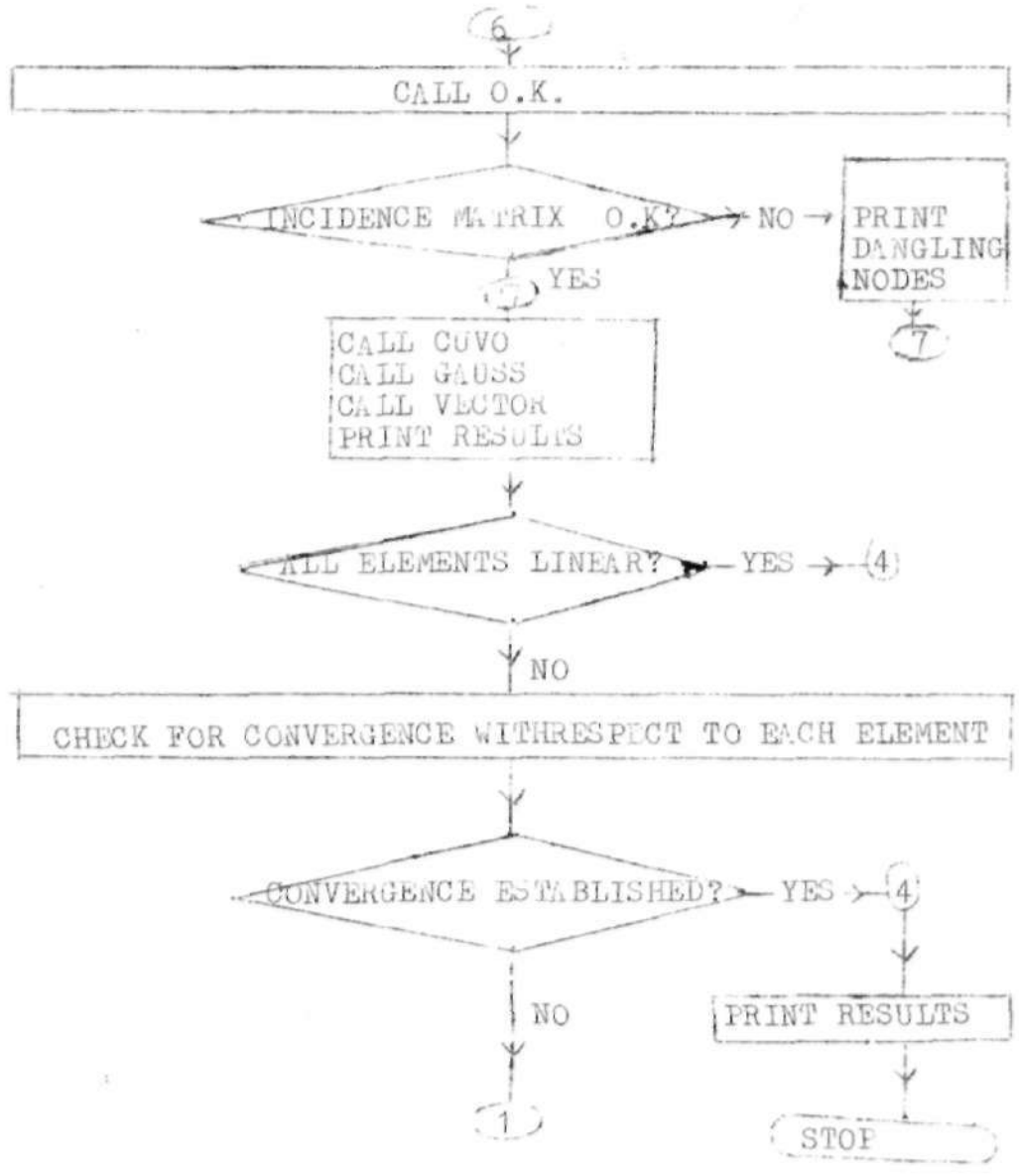


Fig.4.1. Program flow-chart for the General D.C. mode Electronic Circuit Analysis Program.

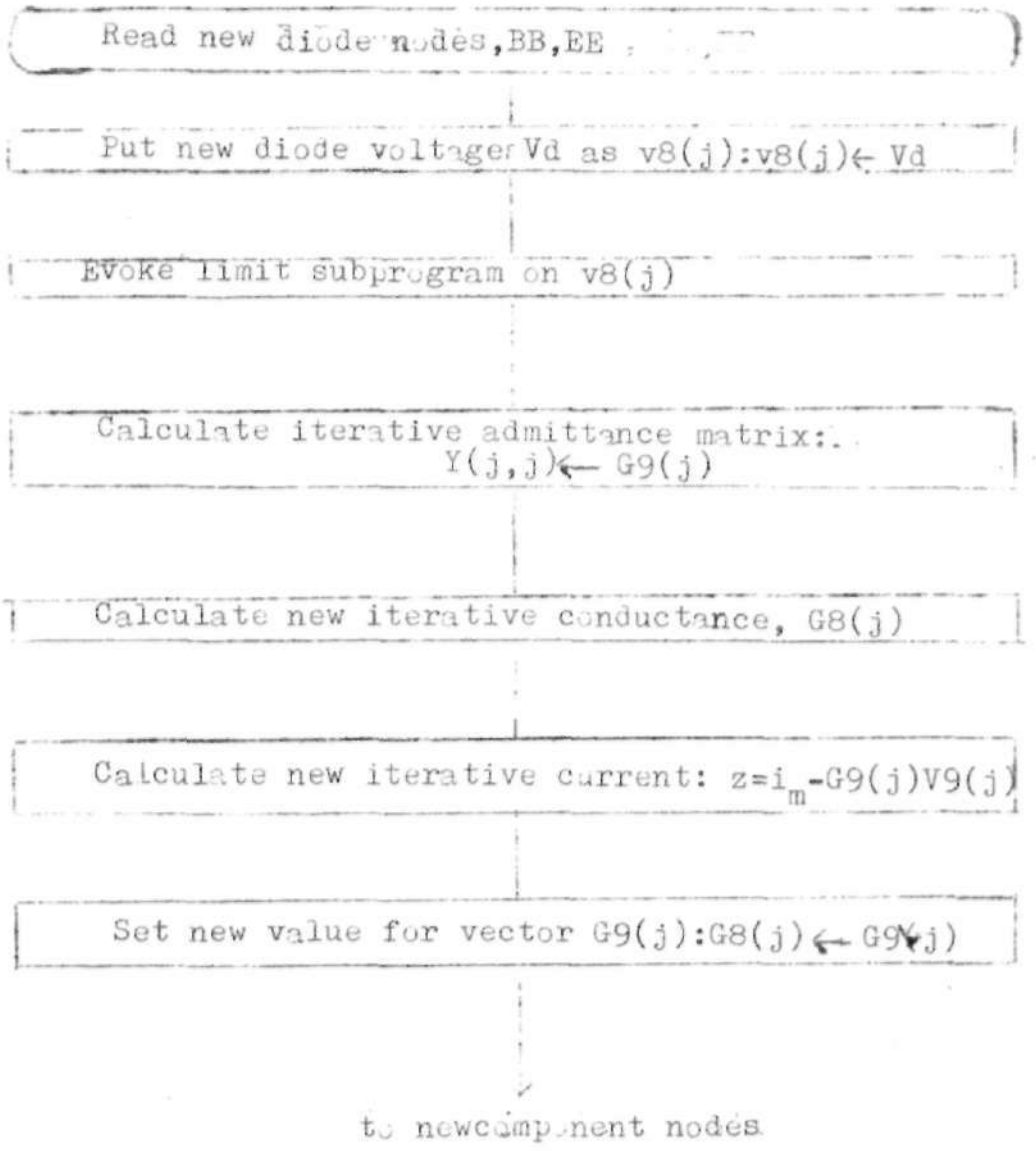


Fig.4.2: Flow chart for the diode modelling routine

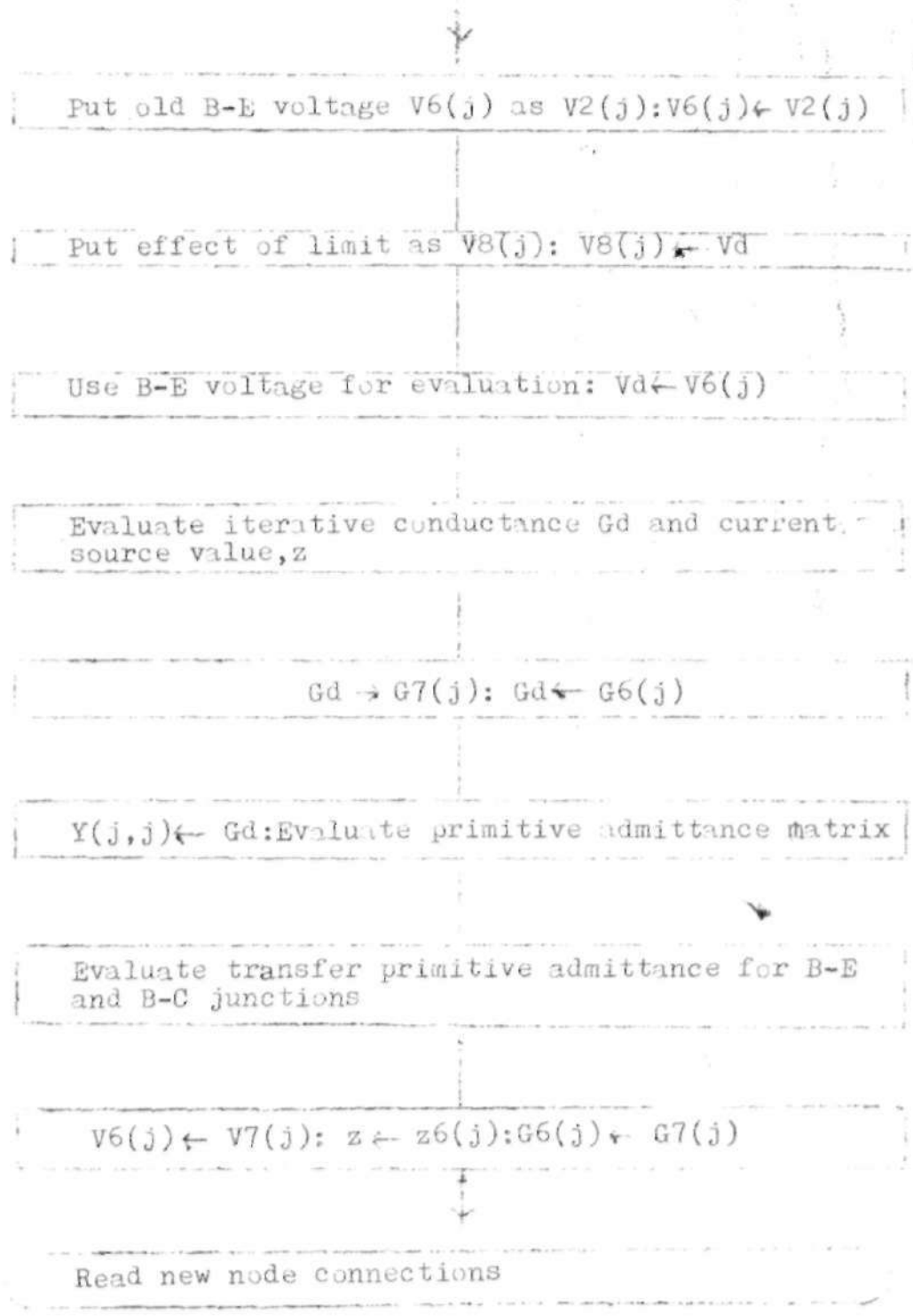


Fig.4.3. Flow chart for the transistor modelling routine.

4.2.4 THE CHANGE ELEMENT VALUE ROUTINE

This routine identifies the element between the nodes required for element value change, and makes necessary changes to the impedance and voltage source value.

The iterative flowchart for the routine is shown in Fig.4.4.

4.2.5 THE DANGLING NODE CHECK ROUTINE

This is the subroutine "OK" in the program. It is used to identify isolated nodes. For each branch, all the nodes are checked for non-zeroes. If the number of non-zeroes is less than two, the message indicating the dangling nodes is printed. The flowchart for the routine is presented in Fig.4.5.

4.2.6 THE ITERATIVE VOLTAGE BOUNDARY ROUTINE

This is the subroutine "LIMIT" in the program. As indicated in Chapter two, placing bounds on possible forward bias diode voltage enhances convergence. As could be seen with the examples presented in Chapter five, very fast convergence, of the iteration is achieved.

The flow-chart for the routine is presented in Fig.4.6.

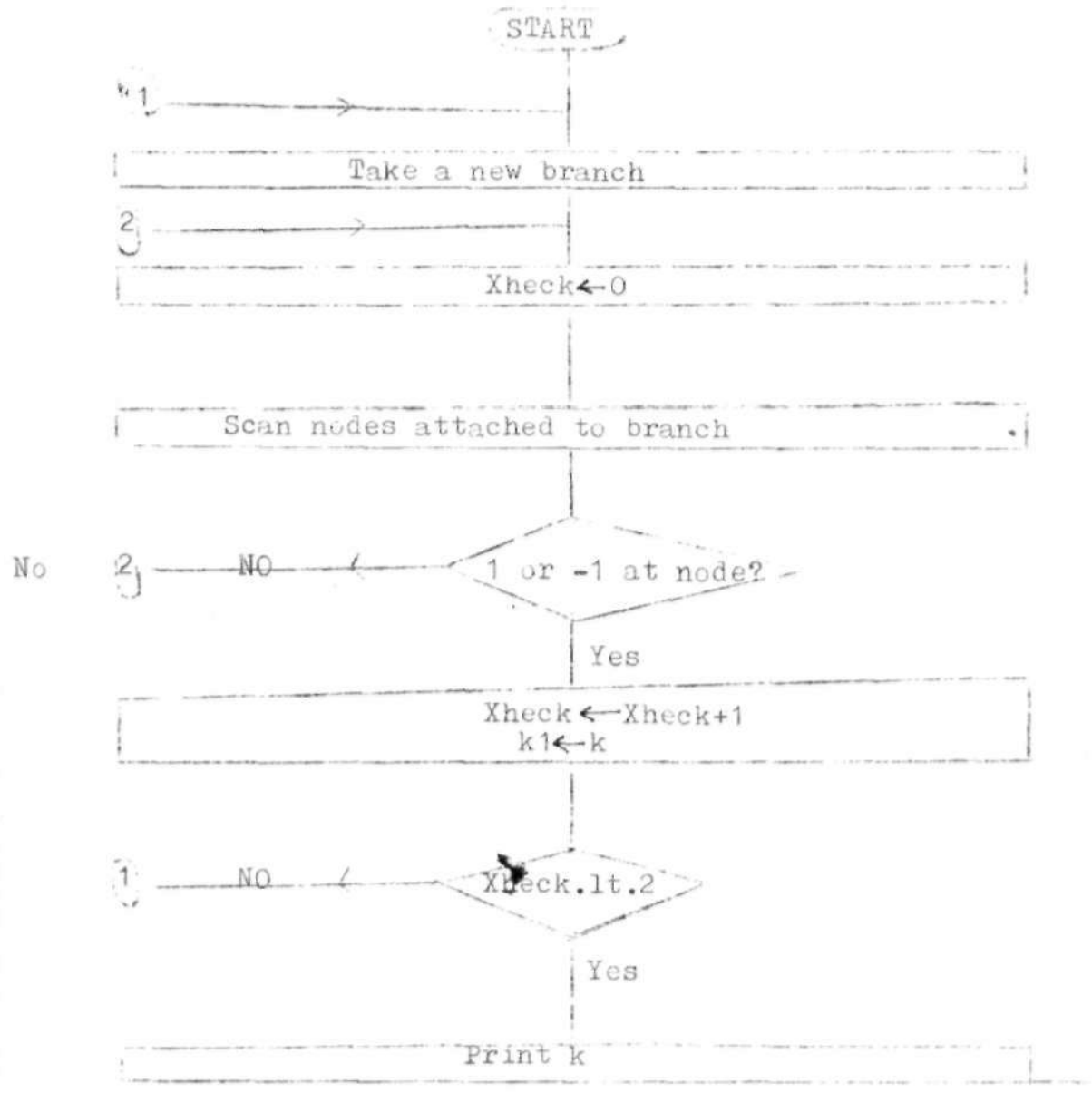


Fig.4.4. Flow chart for the dangling node check routine

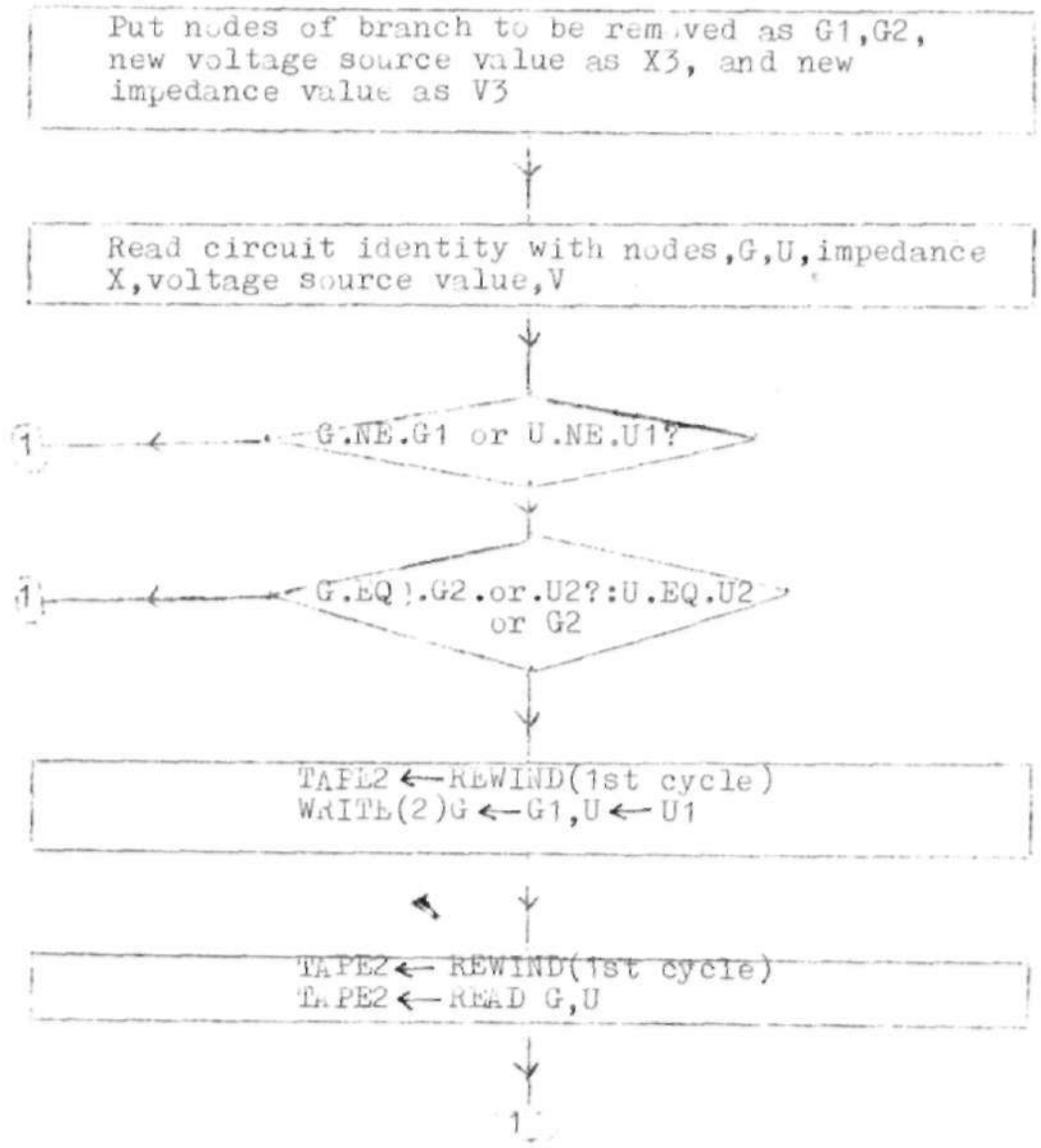


Fig.4.5. Flow-chart for the Change element value routine

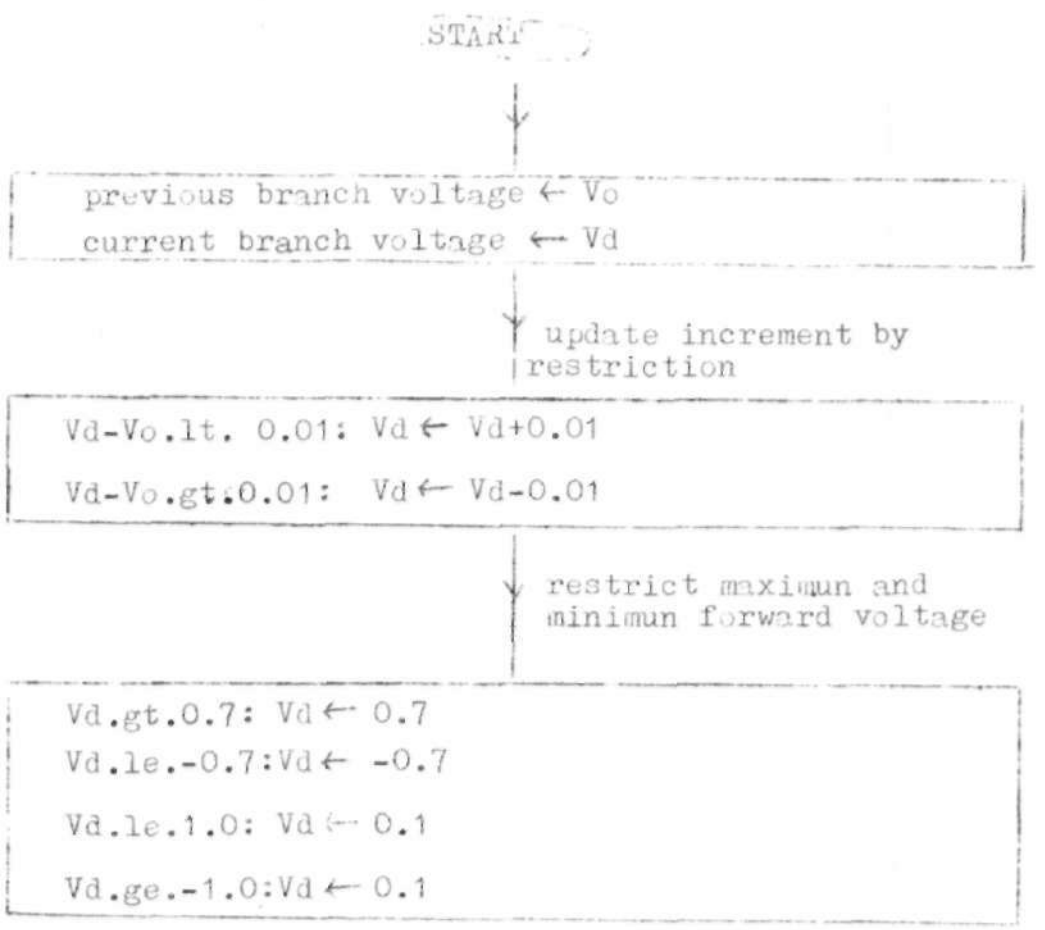


Fig.4.6.Flowchart for the Iterative voltage boundary routine.

4.4 SEQUENTIAL ANALYSIS OF THE INTERACTIVE NATURE OF THE PROGRAM

(i) At log in the user calls the program by its stored name on disc. He may then assign either the printer or the terminal as his output medium.
On running the program, the program will output on the terminal, the source code " INTERACTIVE COMPUTER AIDED DESIGN/ RUMALA, PART, MSC THESIS 1981. AHMADU BELLO UNIVERSITY ". For a first run, it is best to use the terminal as the output medium.
(ii) The prompt message, "DESIGN TITLE" comes next. The title of the analysis work is expected, or a carriage return otherwise.
(iii) Next will come the prompt message, "IS OPERATING TEMPERATURE ROOM TEMPERATURE ?"
If the input response is "YES", the program's next interactive step will be (v).
(iv) A response to (iii) other than "YES" makes the program issue the prompt command " INPUT OPERATING TEMPERATURE ". The expected response to the command is to input the operating temperature in degrees Kelvin.
(v) The next prompt command is "NEW ANALYSIS ?", to which the user is to input a "YES" or "NO" reply.
(vi) The next command issued by the program is "DAMP FACTOR, DPT SOURCE #, TOTAL BRANCH #, TOTAL LINEAR BRANCH #, # OF NODES ".

The expected response is to input the desired iteration damping factor, number of linear dependent sources, total number of branches in the circuit (transistors are counted as having two branches), number of linear resistive branches in the circuit, and the number of circuit nodes in that order.

(vii) The next program message is " CIRCUIT DATA". The expected response is to input all particulars for each element and for all the elements of the circuit. If the element is a resistive element (linear), the expected input is :-

"RES" .....CR

then

NODE1, NODE2, IMPEDANCE, VOLTAGE SOURCE VALUE,

CURRENT SOURCE VALUE .....CR

If the element type is a diode, the expected input is:-

"DIO" .....CR

NODE 1, NODE2 , VOLTAGE RATING .....CR

Zero value is input as the voltage rating for common diodes other than zeners.

If the element type is a transistor, the expected response is :-

"TE".....CR

followed by

COLLECTOR NODE, BASE NODE, EMMITTER NODE, FORWARD VOLTAGE TRANSFER RATIO, REVERSE CURRENT TRANSFER RATIO, Ies, Ics .....CR

The transistor type is input next, say n.p.n.

"NPN".....CR

"NPN" is replaced by "PNP" if the transistor is of the "PNP" type.

If the element type is a FET,

"FET".....CR

is input,followed by

DRAIN NODE, GATE NODE, SOURCE NODE, PINCH-OFF VOLTAGE, TRANSCONDUCTANCE/VOLT RATIO .....CR

followed by the FET type, NFET for an FET operating in the enhancement mode, and PFET for depletion mode operation.

NFET.....CR

The symbol CR indicates a carriage return at the end of the input.

(viii) After completion of the input data, the interactive prompt"WANT TO CHANGE VALUE OR REMOVE A BRANCH?" is displayed. "NO" is input if no element value need changing. "CHANGE" or "REMOVE" is input if an element value needs changing, or an element needs be removed completely from present analysis. In this case, the interactive message "NEW DATA VALUES", followed by "NODE1,NODE2,VOLTAGE,IMPEDANCE" would be displayed.

The expected response is to input the node connections of the element, and the new voltage source and impedance values respectively.

The flow-chart for the over-all procedure is presented in Fig.4.1.

(The FET routine, though developed to some degree in the thesis - due to similarity in model formulation with bipolar transistors, no examples are treated for two reasons : (a) it is not the major concern of this thesis (b) wide spread discrepancies exist in FET parameters).

The computer print-out allows the user to know at a glance which nodes in his circuit he assigned to which component type and make necessary modifications if desired.

(viii) Current flowing through each component, the voltage across it, as well as power dissipated in the device are printed as outputs apart from nodal voltages. This helps the designer/analyst note the power rating of each linear device.

Junction currents and power output in active non-linear devices are to be ignored. For linear voltage sources however, the power and current output at iteration convergence represent the short circuit power and current respectively.

(ix) Program construct allows modification of any one linaer device impedance and voltage source value. Such modifications are stored as local files, except a save command to make such records permanent is given. The nature of the program construct makes the "save" and "replace" commands of the Cyber 72

effective only when the main program is still active.
(x) Program construct allows the user to input circuit elements in any order he feels convenient, i.e. the user may input resistors with thier associated values, followed by diodes , transistors or FETs , or in any mix he feels convenient.
(xi) Program construct allows the user to give his circuit job for analysis a title, e.g. " COLOUR TV POWER SUPPLY DESIGN, 9/10/81" which is then printed as output header , next to the source code.
(xii) The standard input device is used as a means of interactive communication from the user to the computer, and from the computer to the user. (under normal program constructs, a "write on the standard input device error" would occur. This error message is however suppressed due to the nature of the program construct). This enables interactive commands such as "IS OPERATING TEMPERATURE ROOM TEMPERATURE?" to appear only on the terminal, even if a printer or any other output device is assigned as the output medium. This makes for a clean real time interaction since it would be unweidly to have communication meant only for the user at a specified time slot appear in the hard copy printout of results. The program construct thus allows real time interaction at all times irrespective of assignment of output medium. This is desirable as the language in use is in its compiled form rather than interpretive.

4.5 FILE ASSIGNMENTS

Apart from the standard input/output file, three other files are used by the program. Tney are DATA1, DATA2 and DATA8. The first two files are stored in binary tapes 1 and 2 respectively, while the third uses BCD tape 8.
The assignments of data to individual files is done automatically due to the interactive nature of the program construct.
Permanent, saved data are brought to use by the program if "NO" is input to the prompt message " NEW ANALYSIS ?".
New data, which can only be input if "YES" is input to the prompt message "NEW ANALYSIS ?" may be saved to replace old data immediately after the end of the program execution, when the main program is still active, by the use of "REPLACE DATA1, DATA8 " command in the case of Cyber 72 with Minnesotta LevelIV Fortran compiler.

CHAPTER FIVE

APPLICATIONS OF THE ELECTRONIC CIRCUIT ANALYSIS PROGRAM - D.C. MODE ELECTRONIC CIRCUITS

INTRODUCTION

Typical applications of the D.C. mode computer aided design of electronic networks is presented. A study is made of variation of iteration convergence in a particular case of a single transistor amplifier with negative feedback as the feedback resistor is varied in order to test for stability of convergence.

For circuits containing bipolar networks having exponential voltage-current characteristics, the precision of the computer used is very important.

Shier(1960)[17] had indicated that the computing environment(hardware, operating system, and the type of compiler used) influences the result expected in even the simplest of computations. This may be noticeably so for iterative calculations involving exponential functions, where a small change in any iterative element value could propagate as large changes in other components iterative values.

Taking into consideration the degree of floating point precision of the computer used, the two major computational problems are:-

(i) round off errors - Gaussian elimination is especially prone to this;

(ii) quantising errors.

For exponential nonlinearities, the above two errors are sufficient to mar the effect of truncation error used to test for convergence.(9)

A computer with eignt digit floating point precision is certainly of little use in computations of this nature, and if a neccessity arises for using such machine, the program must be run in double precision mode. Circuits with active elements in the feedback loop may require more iteration steps for convergence to be realised.

All-in-all, tne property of tne network is an important determinant as to ease of convergence. Convergence had been obtained for all networks tested so far.
It is suggested however, that large networks be broken down into smaller ones for analysis; this apart from the other precautions such as iteration damping or current source value and resistance value scaling.

The program as written runs on Cyber72 computer, Fortran version 5.0, with the NOS Operating system. The program derives its interactive feasibility by the use of a write on the standard input medium.

The result is a truly interactive program that maintains its interactive nature irrespective of the assignment of the output medium. (The feasibility of this may have to be checked for some other systems.)

Due to the exponential current-voltage relations of bipolar devices, the Newton iteration has a small region of local convergence[10,13]. In this thesis, it had been shown in chapter two that bounding forward exponential voltage changes to within (0.0 to 0.7)volts enhances convergence without restricting the range of the true feasible solution.
Moreover, the user has control over the degree of damping he desires in order to enhance convergence. A lot of trials has impressed the fact that for all circuits tested, a damping factor of between o.1 and no damping at all would suffice. Moreover, the restriction scheme ensures convergence for most bipolar networks , without the need of iteration damping.
Too much damping may result in a prohibitively slow convergence or branch voltage changes not within the range of true solution.

Use is made of the property of exponential and quadractic functions to place a bound. Such bounding was found not to limit the range of true feasible solution - base-emmitter junction voltages less than

zero for a cut-off transistor , between zero and 0.5 for a cutin transistor, and between 0.6 and 1.0 for an active silicon transistor, and greater than 1.0 for a saturated base-emmitter junction of a silicon transistor could still be obtained.
The program construct allows the feed-in of transistor parameters, I_{es}, I_{cs}, α_r, α_f. This is very useful in critical designs, where specific laboratory measurements have to be made for such parameters forindividual transistors.

5.1 TYPICAL INTERACTIVE INPUT SEQUENCE - ANALYSIS OF A VOLTAGE REGULATOR .

A typical input sequence of the program as per Cyber72 is GET,DESIGN (or GET DESIGN, DATA1, DATA8 - for re-executing an old saved circuit data)

ASSIGN, MS = OUTPUT (may be left out for terminal as the output medium)

MNF, I = DESIGN, L = 0

At the end of the compilation, the program outputs

INTERACTIVE COMPUTER AIDED DESIGN / RUMALA, HART, M.Sc. THESIS 1981, AHMADU BELLO UNIVERSITY

(The data input that follows is for the simple voltage regulator of Fig 5.1.
The sign SD implies interactive program screen display, and CR implies the user carriage returns.)
The interactive sequence follows with typical responses.

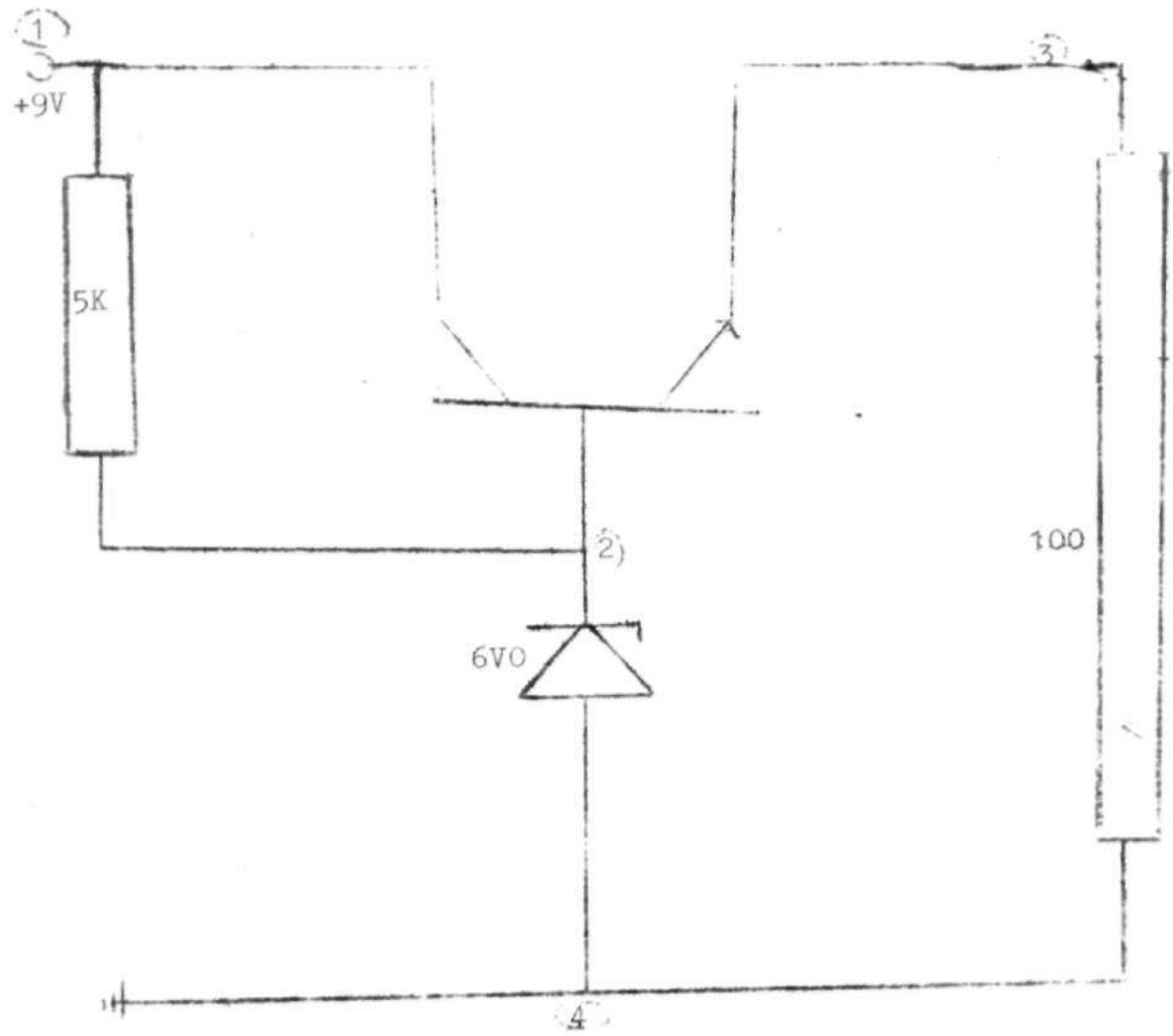


Fig.5.1. Simple series voltage regulator -circuit incorporating zener diode

```
 DESIGN TITLE .....SD
Simple voltage regulator design analysis.....CR
IS OPERATING TEMPERATURE ROOM TEMPERATURE?.....SD
No.....CR
INPUT OPERATING TEMPERATURE..... .....SD
280.....CR
NEW ANALYSIS?.....SD
Yes.....CR
(NO is input for re-executing old analysis)
DAMP FACT R, DPT SOURCE #, TOTAL BRANCH#,
TOTAL LINEAR BRANCH #, # OF NODES.....SD
0.5,0,6,3,4.....CR
CIRCUIT DATA.....SD
Res.....CR
1,2,5e03,0.,0. .....CR
Res.....CR
3,4,3e02,0.,0. .....CR
Res .....CR
1,4, 1e-02,9.,0. .....CR
Dio .....CR
2,4,6.0 .....CR
Te .....CR
1,2,3,.99,.26,4.0e-12,1.9e-10 .....CR
npn .....CR
```

Zeners are input with cathode node first followed by anode node connection. Ordinary diodes are input

with anode node first, followed by cathode node connection,
(The zener model used takes the zener as a diode with
appropriate voltage source in series[2].)
The program will display the data on the terminal or
as hard copy, at this point, depending on the assignment
of the output medium.

WANT TO CHANGE VALUE OF OR REMOVE A BRANCH? .....SD
change .....CR
(CHDIO may be input if zener voltage is to be changed)
NEW DATA VALUES.....SD
NODE1, NODE2, VOLTAGE, IMPEDANCE .....SD
(At this point the program will display the new data
values on the terminal if the terminal is chosen as
the output medium).
The program will iterate from this point until iteration
convergence is established.
/LGO .....SD
Replace, data1..... CR
Replace, data8 ..... CR
Out.....CR
(assuming printer had been assigned prior to program
compilation as the output medium.)
The computer printout for the result of analysis of
circuit of Fig.5.1 is given in Fig.A1,Appendix III.
Node voltages at iteration convergence is given in
Table 5.1.

NODE	VOLTAGE (V)
1	8.995
2	6.2762
3	5.4464
4	GROUND

TABLE 5.1 Table of node voltages for tne simple voltage regulator of Fig.5.1

5.2 LINEAR RESISTIVE NETWORK DEPENDENT SOURCES - OP - AMP TRANSISTOR COMBINATION EXAMPLE

Only voltage controlled current sources are accepted directly by tne program for linear resistive networks. Other linear type of dependent sources may be converted to voltage controlled current source equivalent by use of combination of Thevenin's theorem and Ohmn's law. The sign of the transconductance is taken as positive in tne program construct if the current of the controlled generator is of the same polarity with respect to tne numbering of the nodes as the controlling voltage, and negative otherwise.

Parameters of the controlled sources are requested only if non-zero value is input for the number of dependent sources; whereby the prompt command

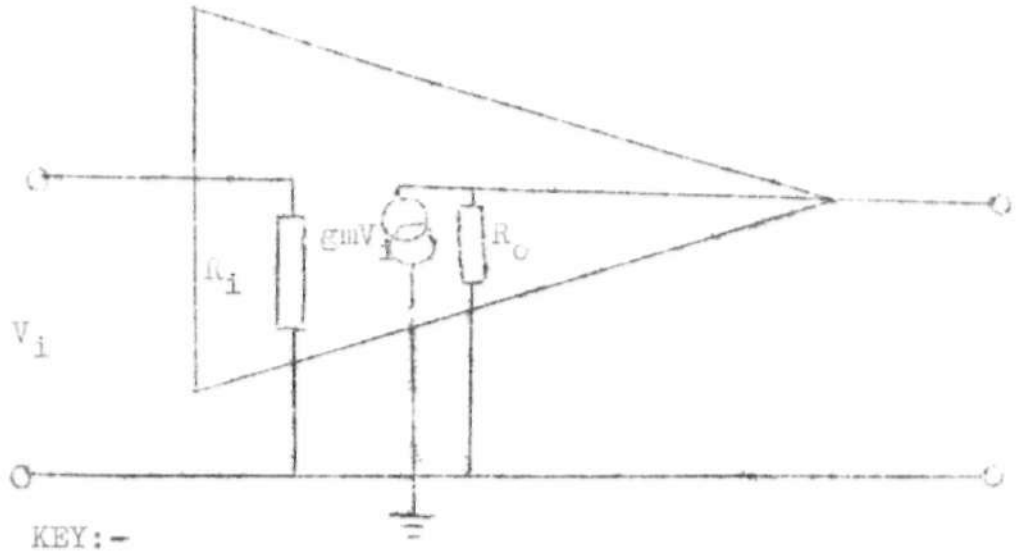
" LINEAR DEPENDENT SOURCES, TO BRANCH, FRO BRANCH,

TRANSFER RATIO" will appear on the terminal. The 'from branch' refers to the branch number of the controlling source, and the 'to branch' to the branch number of the controlled generator.

A good example of a model illustrating a voltage controlled current source is the operational amplifier which may be modelled as shown in Fig.5.2.
An exmaple of a circuit incorporating a voltage controlled current source is given in Fig.5.3.
The circuit has a double-ended supply of 9V; the input voltage to the OP.AMP. is 10mV and the OP.AMP. input impedance is 10Mega-Ohmns, whilst its open loop gain is 100,000.
The result of the analysis of the circuit is shown in Fig. A2, Appendix III.
The computed node voltages at iteration convergence is given in Table 5.2.

NODE	VOLTAGE
1	9.000
2	-8.899
3	-8.890
4	-0.750
5	-8.909
6	9.000
7	GROUND

TABLE 5.2 Node voltages for the linear dependent source - transistor combination example.



KEY:-

V_i = input voltage : R_o = output impedance

R_i = input impedance

Fig.5.2. OP.AMP. model-illustration of dependent sources

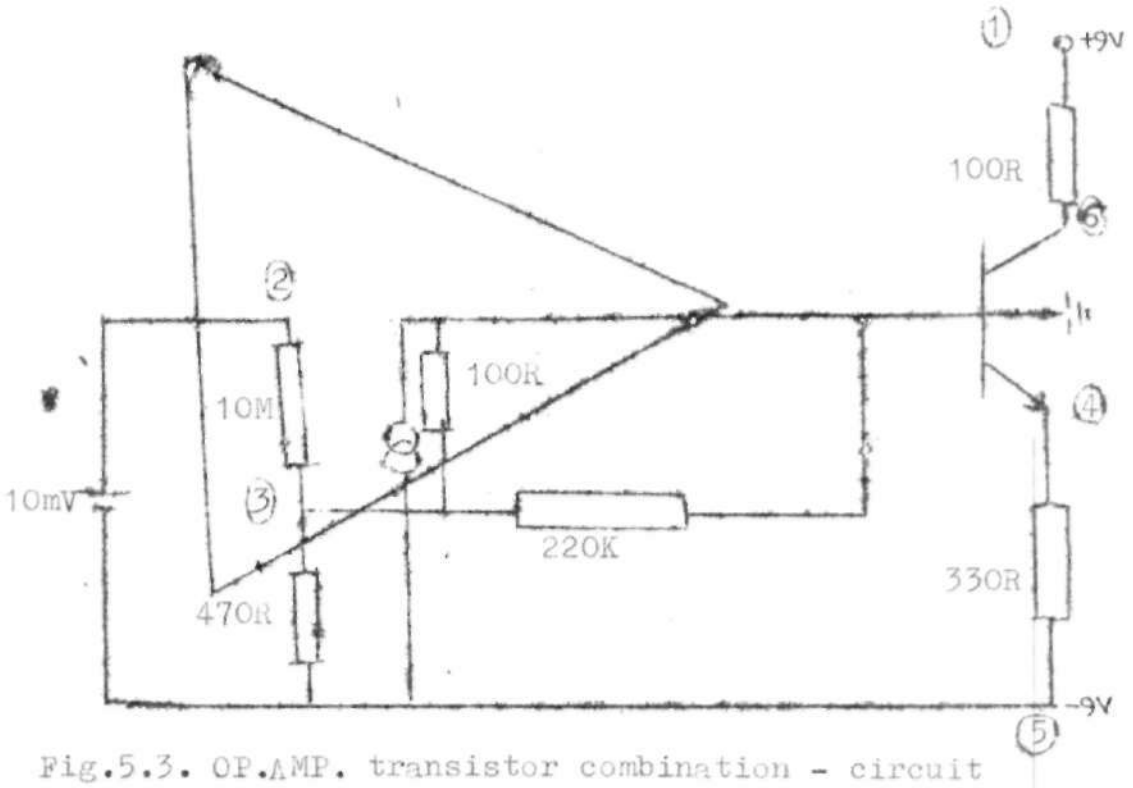


Fig.5.3. OP.AMP. transistor combination - circuit incorporating linear dependent source.

by its internal impedance.

Ideal voltage sources must be approximated by the

5.3 MULTI - STAGE CIRCUIT ANALYSIS : THE SOURCE MOBILITY PROBLEM

From our derivation of nodal circuit analysis in Chapter one, via the standard branch configuration, and noting the standard branch corresponds to a tree branch in a network graph, we note for a given node pair, only one independent current source and one voltage source in series with an impedance is allowed. A few rules are manisiested from above :-

(i) Parallel resistive components should be reduced to their single value equivalent before inputting to the program. This is not a serious restriction for most purely linear resistive networks, and the rule may be flouted in such cases and the correct solution may still be obtained.

(ii) voltage sources specified for a node pair must be in series with each of the parallel combination of the resistive components in order to enhance source mobility. If this rule is flouted for a nonlinear network, iteration convergence may never occur, or a wrong soultion may be obtained.

(iii) pure current sources are allowed between a pair of nodes, but pure voltage sources are not allowed, and each voltage source must be accompanied by its internal impedance.

Ideal voltage sources must be approximated by the

source value in series with a small impedance to satisfy the standard branch configuration outlined in Chapter one.

As an example of the above application, consider the circuit of Fig. 5.4.

Application of the above rules will result in the circuit of Fig5.5.

The result of the analysis of the circuit of Fig5.5 using the program results in output of Fig.A3,AppendixIII. The computed node voltages at iteration convergence is presented in Table 5.3.

NODE	VOLTAGE
1	5.000
2	0.766
3	0.016
4	3.325
5	3.725
6	2.575
7	5.000
8	GROUND

TABLE 5.3 Node voltages for the circuit of Fig5.5. used for the illustration of source mobility.

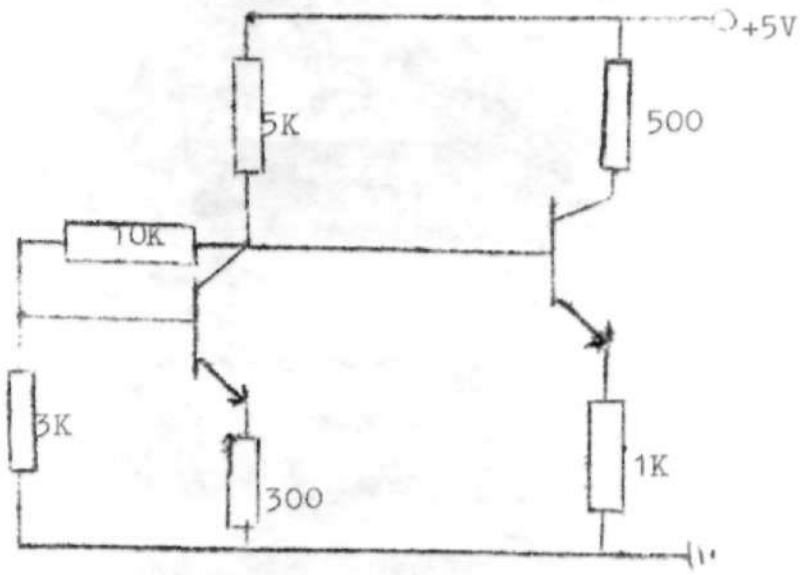


Fig.5.4. Two stage amplifier -source mobility illustration.

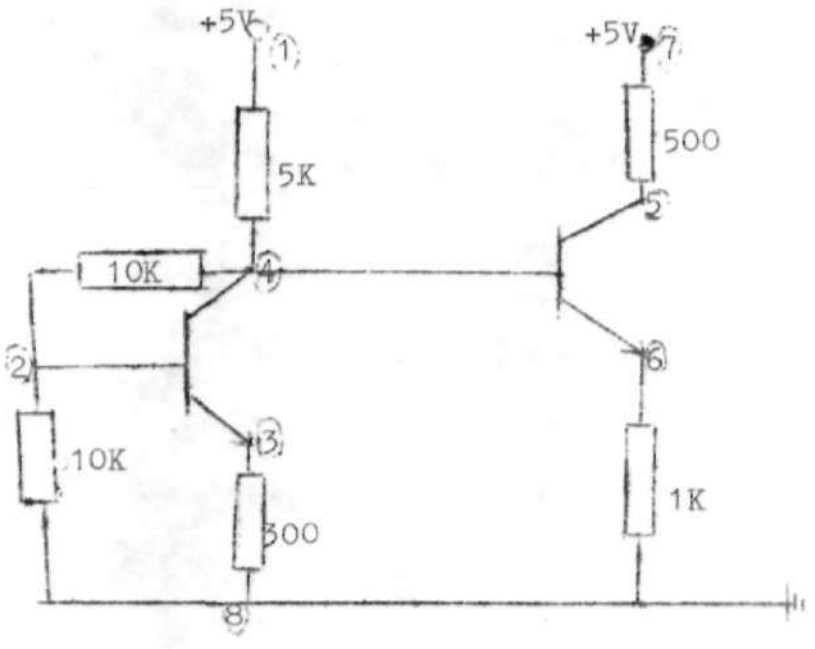


Fig.5.5. Source mobility consideration withrespect to circuit of Fig.5.4.

5.4 VARIATION OF ITERATION CONVERGENCE WITHRESPECT TO MAGNITUDE OF FEEDBACK FOR A SINGLE TRANSISTOR AMPLIFIER

In order to study the stability of convergence withrespect to transistor bias values, the feedback resistor for a single stage amplifier was varied in 1K, 20K, and 30K steps.

The circuit connection is as shown in Fig.5.6.

The feedback resistor connected between nodes 4 and 2 was varied with the program running through the use of the inbuilt "change" command. Typical result of the iterations is presented in Fig. A4, Appendix III. It is noted that for all degrees of feedback, and consequently bias values, the iteration converges at six steps. Hence it may be concluded that for the iterative scheme presented in this thesis, the convergence is highly stable with respect to circuit bias values for a given network - for a node to node feedback.

5.5 THE CIRCUIT TEMPERATURE VARIATION ROUTINE

The program has incorporated into it the circuit ambient operating temperature routine. This is considered a very useful tool since the effect of temperature variation on a circuit is very important to the circuit designer. The inbuilt routine allows the simulation of the ambient working temperature of any circuit so that study of effects such as thermal run-away could be effected.

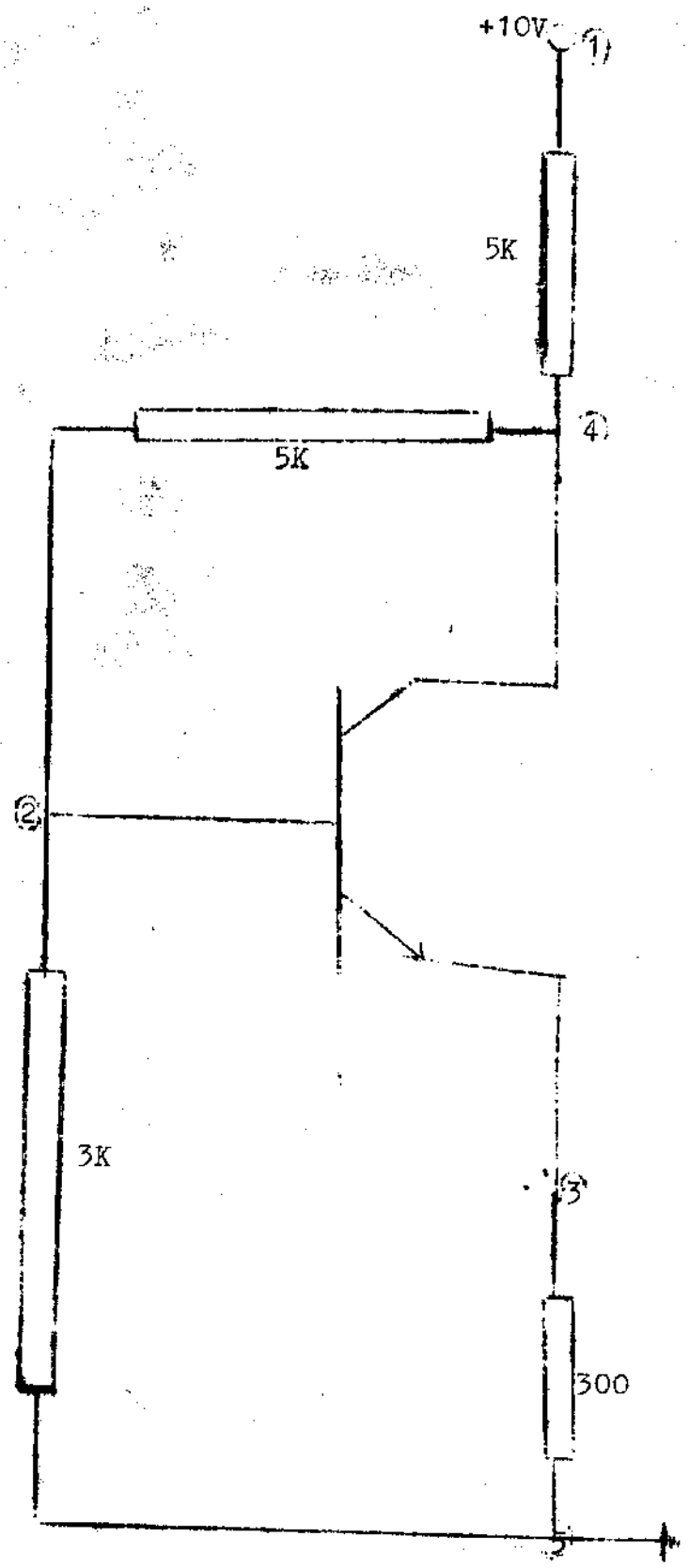


Fig.5.6. Circuit with linear resistive feed-back- demonstration of iteration convergence with magnitude of feedback.

If the analyst wants to change the ambient working temperature of a network, he simply types in "NO" to the prompt message "IS OPERATING TEMPERATURE ROOM TEMPERATURE ?"; whereby the program will give another prompt message "INPUT OPERATING TEMPERATURE ". The operating temperature is given in degrees Kelvin.

The inbuilt routine was tested on the PNP transistor amplifier network given in Fig. 5.13, to simulate working temperatures at 200°K, 250°K, 300°K, and 400°K. The results of the computer analysis is as tabulated in Tables 5.4 and print-outs are as in Fig. A5, Appendix III,

TEMPERATURE (°K)	NODE VOLTAGES(V)			
	NODE1	NODE2	NODE3	NODE4
200	5.000	1.829	2.497	2.197
250	5.000	2.582	3.232	1.460
300	5.000	1.858	2.509	2.183
350	5.000	1.984	2.631	2.060
400	5.000	2.431	3.692	1.101

Table 5.4 Variation of node voltages with temperature for circuit of Fig.5.7.

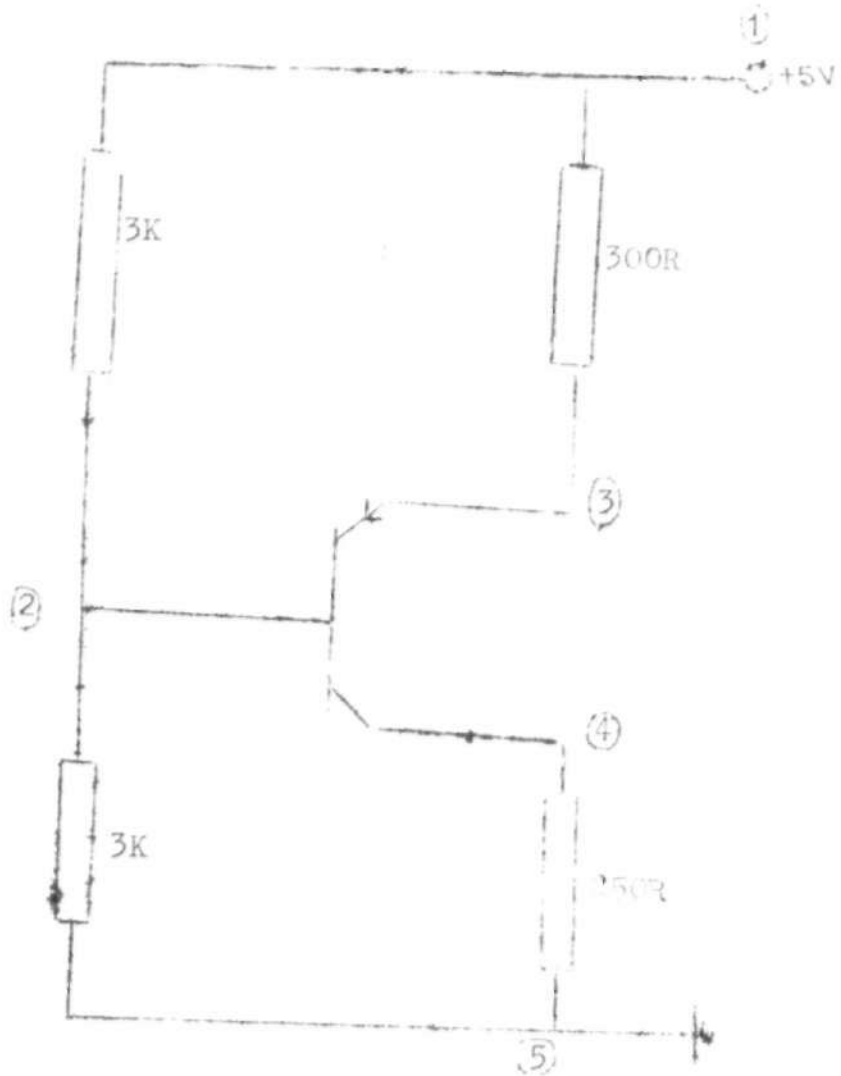


Fig.5.7. Circuit used to test the temperature variation routine.

5.6 ANALYSIS OF A STANDARD VOLTAGE REGULATOR - CIRCUIT WITH ACTIVE COMPONENT IN FEED-BACK LOOP

In order to study the behaviour of the program to active components in the feed-back loop, a standard voltage regulator with circuit diagram as in Fig. 5.8 was analysed.

The circuit redrawn for iterative analysis is given in Fig. 5.9.

With node 7 taken as the ground node, the results of the computer analysis via the program is as tabulated in Table 5.5. An iterative damping factor of 0.5 was used for the analysis.

LOAD(R_L) (Ohms)	NODE VOLTAGES						LOAD	
	1	2	3	4	5	6	V_L(V)	I_{RL}(A)
00	12	8.697	7.947	6.150	6.900	12	0	0
1K	12	8.697	7.947	6.150	6.900	12	7.917	0.008
500	12	8.696	7.946	6.150	6.900	12	7.946	0.016
200	12	8.695	7.944	6.150	6.900	12	7.944	0.040
100	12	8.692	7.941	6.150	6.900	12	7.941	0.079
50	12	8.687	7.935	6.150	6.900	12	7.935	0.159
10	12	8.649	7.887	6.150	6.900	11.997	7.887	0.789
1	12	8.242	7.388	6.150	6.899	11.92	7.388	7.388

TABLE 5.5 Node voltages variations with load for circuit of Fig.5.9



Fig.5.8. Circuits with non-linear ,transistor, feedback-a voltage regulator.

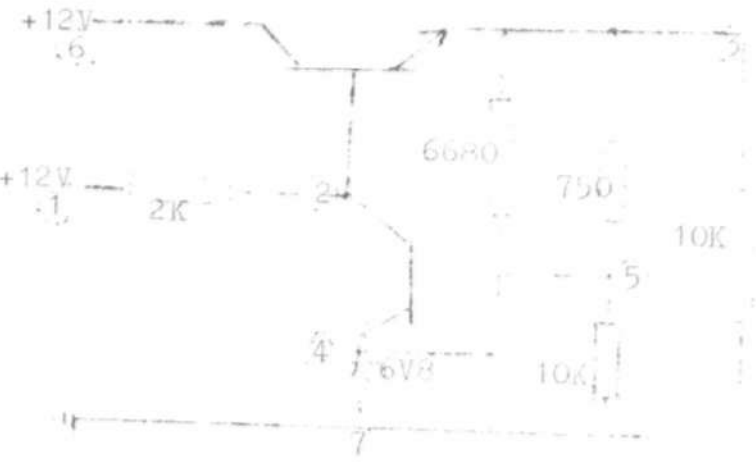


Fig.5.9 . The voltage regulator of Fig.5. redrawn for iterative analysis.

A typical output sequence is given in Fig.A7, App. diIII. Convergence was achieved for all degrees of loading, despite the fact that an active element , the resistor, formed part of the feedback loop.

5.7 INCREMENTAL TRANSISTOR MODELS AND SMALL SIGNAL VOLTAGE GAIN CALCULATIONS (7).

Consider the common emmitter cascade of Fig.5.10, with the transistors biased in the active region. Assume signal source amplitude of 1mV with source impedance of 10Kilo-Ohms. A pi-model representation of the circuit is given in Fig.5.11.
If the bias conditions in the circuit are such that $g_{\pi 1} = 0.1$ and $g_{\pi 2} = 0.2$ (this could have been evaluated by feeding the appropriate circuit data into the program, noting that $g_m = 40I_e$, where I_e is the emmitter current), we get node voltages as per Table 5.6.

NODE	VOLTAGE(V)
1	0.000
2	0.000
3	-0.004
4	-0.004
5	0.190

TABLE 5.6 Node voltages for the circuit of Fig.5.11.

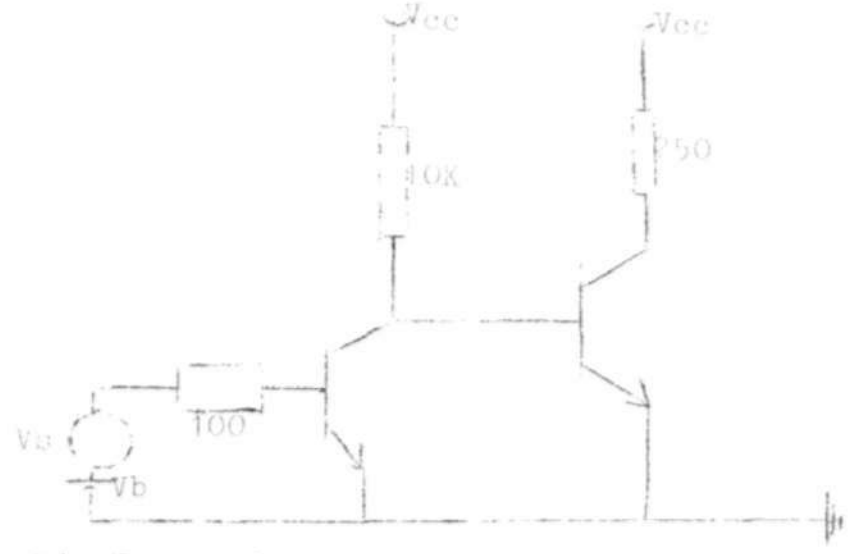
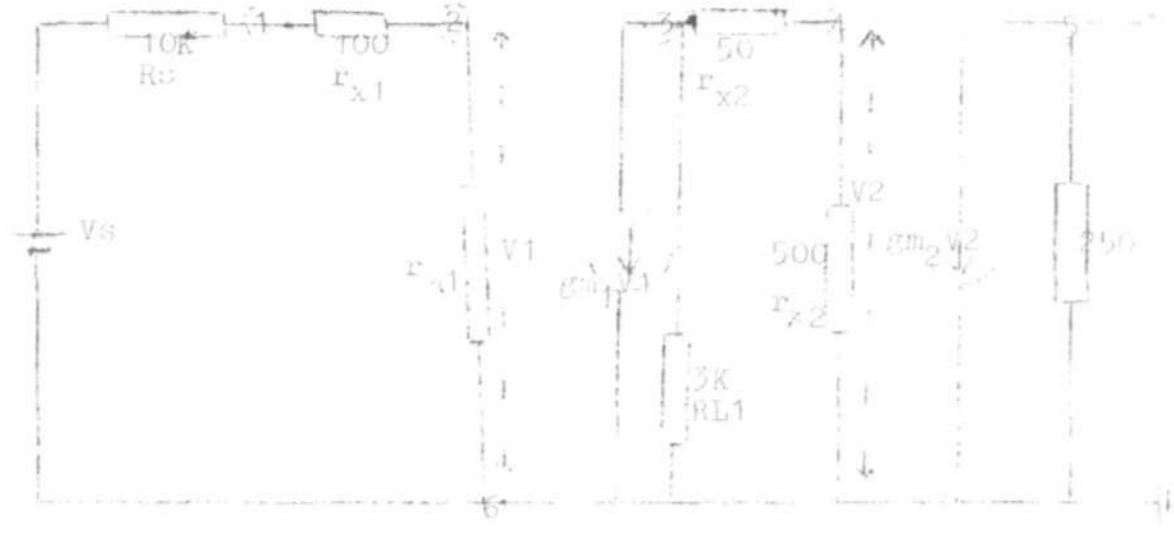


Fig.5.10. Circuit used to illustrate small signal voltage gain calculations via the pi transistor model(Vs=signal source amplitude,Vb=bias voltage)



KEY:-

Vs=signal source amplitude;r_{π}=forward bias B-E junction resistance; r_x=base spreading resistance gm=incremental input resistance

Fig.5.11. pi-model representation of circuit of Fig.5.10.

Taking the earth as the reference point, the voltage gain at node 3 = -0.004/0.001 = -4
voltage gain at node 5 is 0.190/0.001 = 190
The program by itself thus suffices for the evaluation of D.C. mode voltage gain evaluation of a bipolar transistor network.

5.8 THE CUT -OFF TRANSISTOR ANOMALY - ANALYSIS OF A VOLTAGE REGULATOR INCORPORATING A CURRENT LIMITTER

Circuits containing cut-off transistors may show marked anomaly of the iterative voltages at convergence at nodes which the cut-off transistor has effect.
The anomaly is marked for transistor for which the base current is insufficient to turn on the transistor or for transistors with excessive collector loading.
A typical case is presented for the voltage regulator of Fig. 5.12 which incorporates a current limitter.
The circuit redrawn for iterative analysis is shown in Fig. 5.13.
Using α_F = 0.99 and α_r =0.26 and I_{cs} =4.9 x 10^{-12}A, I_{cs} = 1.9 x 10^{-10}A results in the node voltages presented in Table 5.7.

	LOAD (KILO - OHMS)							
NODE	5	1	0.5	0.2	0.1	0.05	0.03	0.001
1	NODAL VOLTAGES (V)							
1	12	12	12	12	11.999	11.998	11.998	12
2	12	12	12	12	12	12	12	12
3	11.796	9.952	9.844	9,980	10.205	11.406	11.406	46
4	10.631	8.997	8.886	9.012	9.223	9.692	10.350	42.5
5	8.909	7.503	7.417	7.547	7.763	8.236	8.836	40.1
6	6.401	6.425	6.467	6.597	6.813	7.287	7.947	41.1
7	0.051	0.075	0.117	0.247	0.463	0.936	1.483	35.2
V_{RL}	10.630	8.922	8.769	8.765	8.760	8.795	8.862	

TABLE 5.7 Node voltage variations with load for circuit of Fig 5.13 - illustration of cut-off transistor anomaly.

A study of the table leads to the following deductions :-

(i) with light loading, 5K,load resistance, transistor T2, and T1 are inactive due to insufficient base-emmitter current flowing through them. The voltage at node 6 retains the bias value for the zener, while the voltage at node 5 retains the potential divider value. The voltage at node 4 overshoots its design value.

(ii) with load values greater than 50 Ohms but less than 1K Ohms, T2 is active and regulation effect is seen from the voltage, V_{RL}, across the load.

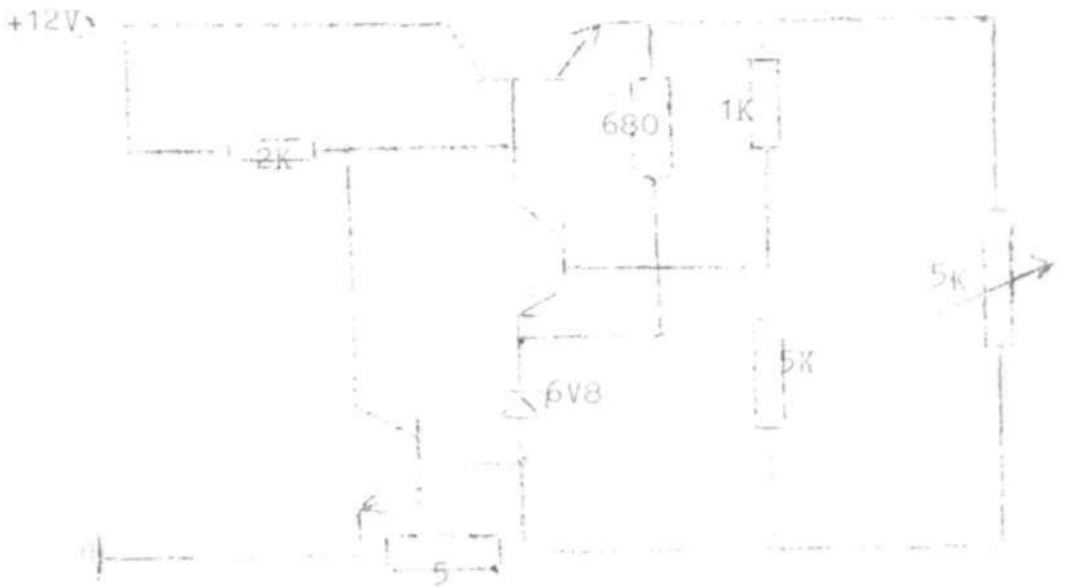


Fig.5.12. Voltage regulator incorporating a current limitter-illustration of the cut-off transistor anomaly.

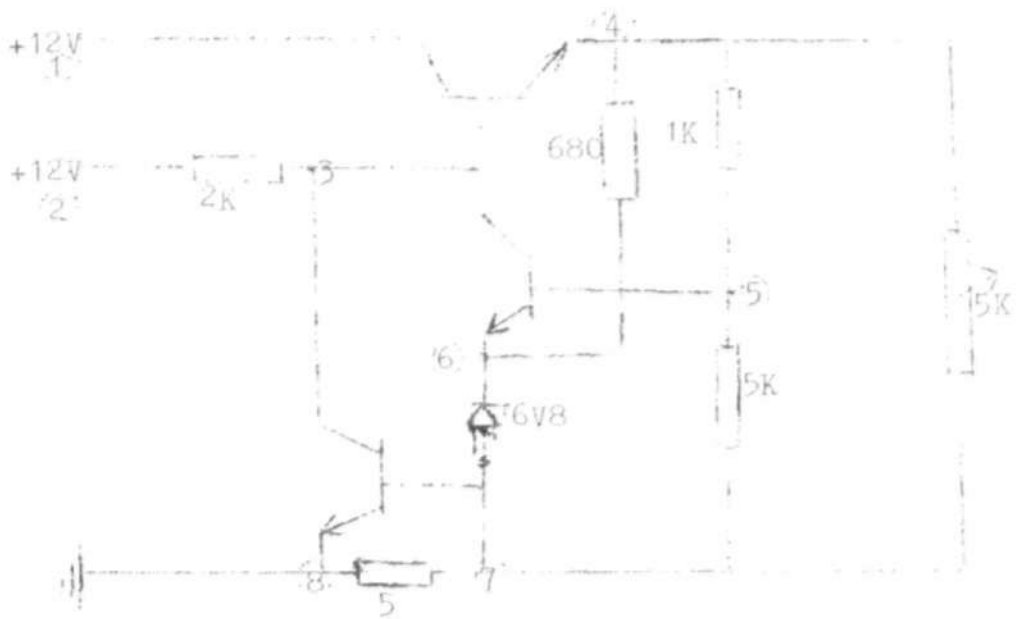


Fig.5.13. The circuit of Fig.5.24 re-drawn t. for circuit analysis

(iii) with load values greater than 50 ohms, the cut-off transistor usually is felt at nodes 3,4,5,6, 7, the anomaly becoming more pronounced as the loading increases - the result at 1 Ohm loading, for example.

It may be concluded that the result when there is insufficient base emitter current to bias a transistor is authentic, but the result when a transistor is overloaded is anomalous. With proper iteration damping, no confusion can occur in the interpretation of the results so obtained, since the effect of an overloaded transistor on the results at convergence is obviously noticed and the overloaded transistor easily pin-pointed.

5.9 COMPARISON OF THE THESIS PROGRAM WITH THE MOST WELL KNOWN D.C. MODE NETWORK ANALYSIS PROGRAMS

The attributes of the program researched in this thesis is better appreciated if it is compared with the most popular electronic network analysis programs researched in various Universities and major Computer Institutions.

Table 5.8 gives the appropriate comparison.

FUNCTION	PROGRAM TYPE				
	THESIS	(9) IBM ECAP	(2) POTTLE	(2) CIRCUS	(2) CSUCAP
LINEAR ANALYSIS	*	*	*		*
NON LINEAR ANALYSIS	*	*		*	*
FORMULATION	Nodal	Nodal	State Variable	State Variable	State Variable
D.C. Sensitivity	via change command	*			
Operation	on-line	on-line	batch	batch	on-line

TABLE 5.6 Comparison of the thesis program functions with other well-known ECAP programs

The thesis program thus has most of the attributes of the most well - known programs, save A.C. nodal analysis which is similar to the linear part of the thesis program but for the fact that sources and impedances are complex in nature. With little change in formulation, A.C. nodal analysis could easily be accomodated into the program. The thesis concern lies on the more difficult non-linear D.C. mode design, though linear D.C. mode design is also accomodated.

CHAPTER SIX

CONCLUSION

Of importance also, is the fact that the user does not need to guess the initial condition, the initial starting value being internally fixed by the program.

Dedicated methods such as sparse matrix techniques based on L-U factorisation schemes were not used but are in non-linear computations, the elements of the nodal admittance matrix change between solutions, and if such a scheme were used, there would be need to make sure that the whole admittance matrix is not refactored when only part of it changes, or else the purpose of such a scheme over the Gaussian elimination method would be defeated. (Program construct gives allowance for virtually any method of solving linear equations to be used).

Like all circuit analysis programs, restrictions as to the type of non-linear networks that could be analysed is unavoidable.
Sources of restriction is noticeable if Ohms law is expressed in its hybrid form, viz:-

$$\begin{bmatrix} i \\ v \end{bmatrix} = \begin{bmatrix} G & C \\ P & R \end{bmatrix} \begin{bmatrix} V_c \\ I_c \end{bmatrix} = \underline{H}\,\underline{w}$$

where R represents the transresistances or CCVS, G, the transconductances or VCCS, C the transfluences or CCCS, and P the transpotentials or VCVS

6.1 AREAS OF FURTHER RESEARCH

An improvement to the program would be for the program to be able to tackle all the four transfluences that occur in the expression of Ohms law in its hybrid form whenever they are encountered in a circuit, and to allow all the four transfluences to be non-linear.
For the program to be complete, the problem of dependence of the $\underline{z}$ vector on the $\underline{w}$ vector in the expression of Ohms law in its hybrid form, has to be tackled. These are highly complicated issues the characteristics of which are yet to be covered by any one known single DCAP program.

Solution of the non-linear equations effected via linearisation. New methods of solving the non-linear equations directly would be effected with advancing research. A scheme not tried could be based on the "Fortran subroutine for solving systems of non-linear simultaneous equations", Incerti and Zirrilli[9](1981).

Though sufficiently comprehensive treatment was given to the bipolar modelling, the FET model needs further research, especially as there exists different types of FETs, and even for the same FET the characteristics differ with structuring. The thesis therefore, like the CoBOrD code by Calahan(1972), included no FET examples. The modelling as well as the programming would however prove to be quite a further research work.

Finally, it is appreciated that numerical solution of non-linear network problems can be subject to errors of their own that could form basis of further investigations, e.g.:-

(i) the dependence of the convergence to the true solution on circuit parameters

(ii) the dependence of convergence to the true solution on the degree of coupling

(iii) the dependence of convergence to the true solution on linearisation error of the original non-linear equations

(iv) the dependence of convergence to the true solution on computer round-off errors which is related to the numerical accuracy of the computer used as well as the circuit topology - certain circuit topology causing less round-off errors than others. Here - over such round-off errors may cause numerical instability for certain circuit topology.

Thus further research work would have to be done to ascertain the universality of convergence with respect to circuit topology and the other conditions stated above.

All standard circuits tested with the numerical procedures of the thesis have indicated that convergence implied obtainability of the true circuit solution. This is an indication of the feasibility of the solution method.

On the whole, the computer aided circuit design program is an invaluable tool for the circuit designer that could save time and cost in prototype design; the solution being available in seconds instead of in days or months as may be the case with laboratory breadboarding.

The program researched in this thesis incorporates a number of features normally not found in conventional non-linear D.C. analysis program, such as ability to remove a component or null its effect; a dangling node check routine that serves as an indication of dangling components; ability to change the value of a component, and variable transistor parameter capability as opposed to fixed models - all within a compact program. These features will even enhance a faster design turn-around of completed designs as compared to some other well known design programs.

APPENDIX III

PROGRAM LISTINGS OF MODELS OF ACTIVE DEVICES

Program data listings, Figs A1 to A4, are the results of the iterative analysis, each shown listed in the listing(or vision seen in the preceding titling).

Since the analysis is modal, node voltages are printed with respect to the highest number node, taken as the reference or ground node.

Branch voltages(transistors are referred to as being two branches, B-E branch and C-E branch), however, are evaluated as well. The current flowing through a branch as well as the power dissipation in the branch are evaluated as well.

No difficulty could arise in interpreting the result with regard to active elements, as outlined in example in Chapter five.

INTERACTIVE COMPUTER AIDED DESIGN/RUNALA,PART,MSC.THESIS 1981,AHMADU BEL

SIMPLE VOLTAGE REGULATOR DESIGN

TYPE	NODE1	NODE2		OHMS	VOLT	CURRENT
RES	1	2		.500E+04	0	0
RES	3	4		.100E+03	0	0
RES	1	4		.100E-01	9.0000	0
DIO	4	2	6.0000			
TE	1	2	3	.299E	.2500	.49EE-11 .19LE-09 NPN

NODE	VOLTAGE
1	9.0000
1	9.0000
2	0

BRANCH	VOLTAGE	CURRENT	POWER
1	E.0000		
2		0	0
3	9.0000		
4	E9.1EEE	0	0
5	E9.1EE0	0	0
6	-.0000	0	0

NODE	VOLTAGE
1	9.0000
2	8.9998
3	.0000

BRANCH	VOLTAGE	CURRENT	POWER
1	.0002		
2	.0000	.0000	.0000
3	9.0000		
4	E8.9998	.0000	
5	E8.9998	-.0000	.0000
6	.0002	.1EE0	

NODE	VOLTAGE
1	9.0000
2	8.9895
3	.0000

BRANCH	VOLTAGE	CURRENT	POWER
1	.0105		.0000
2	.0100	.0000	
3	9.0000		
4	E8.9895	.0000	-.0000
5	E8.9895	-.0000	.0000
6	.0105	.0000	.0000

NODE	VOLTAGE
1	8.9994
2	6.0431
3	5.9129

BRANCH	VOLTAGE	CURRENT	POWER
1	2.9563	[illegible]6	.0017
2	5.9129	.1591	.3496
3	8.9994	*	*
4	=6.1431	-.1[illegible]2	.[illegible]
5	-.1312	-2.8938	.3768
6	2.9563	2.8649	8.4693

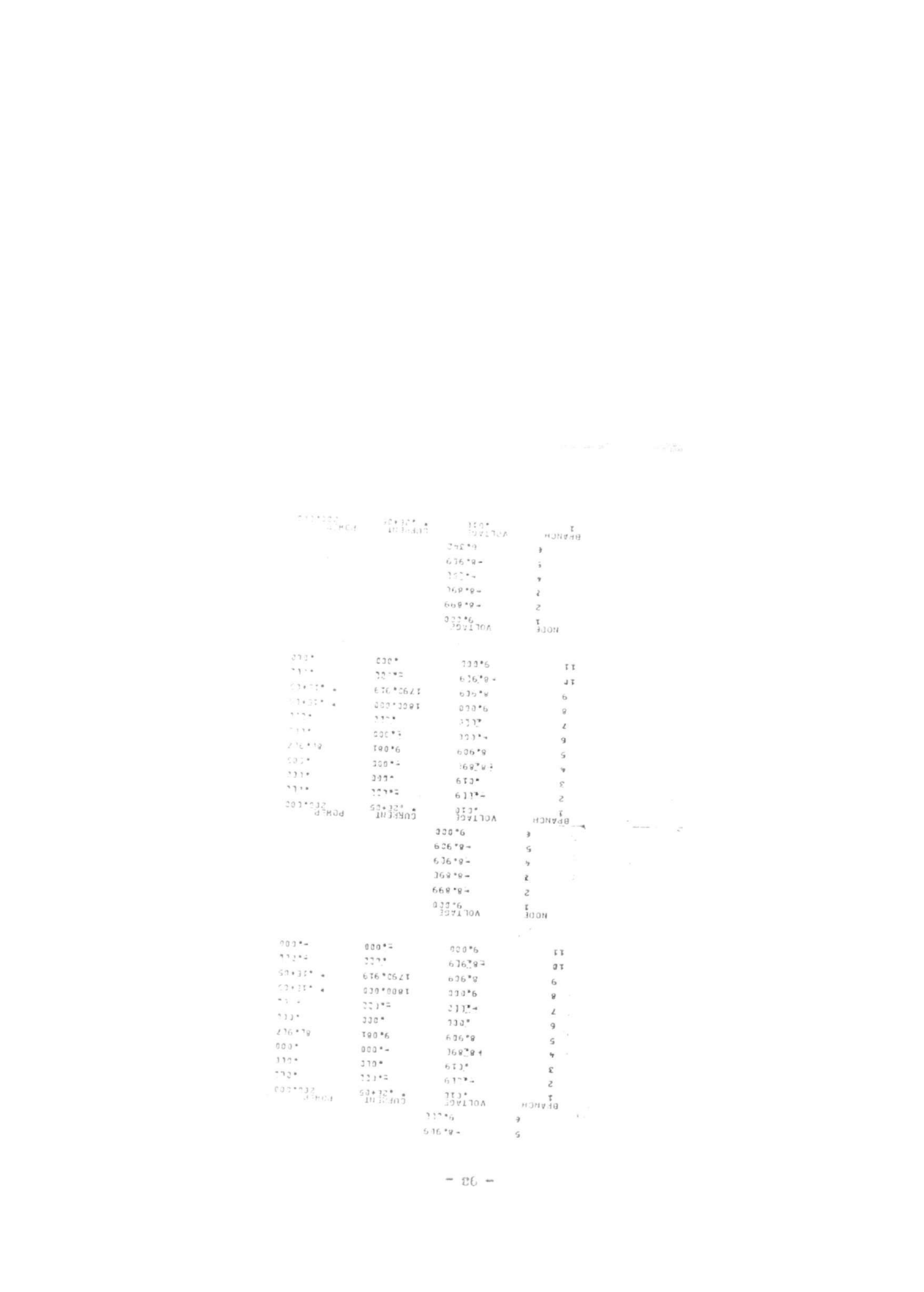
NODE	VOLTAGE
1	8.9995
2	6.2762
3	5.4464

BRANCH	VOLTAGE	CURRENT	POWER
1	2.7232	[illegible]5	.0015
2	5.4464	.0545	.2966
3	8.9995	*	*
4	=6.2762	-.2000	.0011
5	-.8298	*	15.5698
6	2.7232	18.5756	50.5855

NODE	VOLTAGE
1	8.9995
2	6.2762
3	5.4464

BRANCH	VOLTAGE	CURRENT	POWER
1	2.7232	.0005	.0015
2	5.4464	.0545	.2966
3	8.9995	*	*
4	-6.2762	-.0000	.0000
5	-.8298	*	15.5698
6	2.7232	18.5756	50.5854

FIG. A2



5	3.965	3.181	8.961
6	-8.159	-.125	.212
7	2.448	.024	.063
8	9.000	1797.975	* .1E+15
9	8.909	1791.834	* .1E+15
10	.751	-15.229	79.781
11	6.552	100.261	889.691

FIG. A4

```
INTERACTIVE COMPUTER AIDED DESIGN/PUMAL+,PART,NSC      THESIS 1961,AUGUSTO A-L30 3X2

TWO-STAGE AMPLIFIER- CURRENT SOURCE MOBILITY ILLUSTRATION

                CIRCUIT DATA
TYPE   NODE1  NODE2      OHMS          VOLT          CURRENT
RES      2      8      .2[illegible]E+[illegible]      5.[illegible]          1
RES      1      4      .500E+14          0             0
RES      4      0      .100E+05          0             0
RES      2      8      .3[illegible]E+[illegible]          0             0
RES      3      8      .300E+[illegible]          0             0
RES      6      8      .2[illegible]E+[illegible]          0             0
RES      7      5      .5[illegible]E+[illegible]          0             0
RES      7      8      .100E+01       5.000          0
TE       4      2      3   .9900   .[illegible]   .49[illegible]E-11   .29E-09   NPN
TE       5      4      6   .9900   .2600   .4900E-11   .29E-09   NPN

NODE      VOLTAGE
  1         5.000
  2          .833
  3            0
  4         3.611
  5         5.000
  6
  7         5.[illegible]

BRANCH    VOLTAGE      CURRENT      POWER
  1       5.[illegible]      [illegible].[illegible]      4399.995
  2       2.369          .333          .000
  3       2.778          .[illegible]          .[illegible]
  4        .833          .000          .000
  5          .             .             0
  6          0             0             0
  7          0             0             0
  8       5.[illegible]      [illegible].[illegible]      5.[illegible]
  9        .[illegible]             .             .
 10       2.778             .             .
 11      -3.611             0             0
 12       2.[illegible]             .             0

NODE      VOLTAGE
  1         5.000
  2          .833
  3        -3.[illegible]
  4         3.611
  5         5.000
```

2812x7

1	5.000	999.999	4999.988
2	3.041	.001	.013
3	.918	.000	.010
4	.252	.000	.000
5	.202	.001	.003
6	.219	.001	.004
7	.549	.001	.001
8	5.000	999.999	4999.984
9	-.150	-37.587	.354
10	.918	7.015	6.368
11	-.050	-37.588	.354
12	3.291	7.017	23.096

NODE	VOLTAGE
1	5.000
2	.766
3	.016
4	3.325
5	3.725
6	2.575
7	5.000

BRANCH	VOLTAGE	CURRENT	POWER
1	5.000	1000.000	4999.995
2	1.675	.000	.001
3	1.559	.000	.001
4	.766	.000	.000
5	.016	.000	.000
6	2.575	.013	.033
7	1.275	.003	.003
8	5.000	999.997	4999.962
9	-.751	-21.063	79.725
10	1.559	115.237	269.325
11	-.750	-106.302	79.731
12	.401	105.239	42.077

NODE	VOLTAGE
1	5.000
2	.766
3	.016
4	3.325
5	3.725
6	2.575
7	5.000

BRANCH	VOLTAGE	CURRENT	POWER
1	5.000	1000.000	4999.995
2	1.675	.000	.001
3	1.559	.000	.001

4		.766	.003	.000
5		.826	.000	.000
6		2.575	...3	...7
7		1.275	.003	.003
8		9....	999.907	4999.962
9		-.795	-1[illegible].3[illegible]	79.725
10		2.559	105.237	269.325
11		-.751	-1[illegible].3[illegible]	79.73[illegible]
12		.486	215.239	42.[illegible]77

FIG.A4

APPLICATION OF THE 'CHANGE' COMMAND

INTERACTIVE COMPUTER AIDED DESIGN/ANALYSIS,PART,MOD

EFFECT OF FEEDBACK ON CONVERGENCE

TYPE	CIRCUIT NODE 1	NODE 2	DATA	VOLT	CURRENT
RES	1	5	.1000E-01	10.000	0
RES	4	2	.2000E+05	0	0
RES	2	5	.3000E+04	0	0
RES	3	5	.3000E+03	0	0
RES	1	4	.5000E+04	0	0
TE	4	2	3 .9900	.2600	.4900E-11 .1900E-09 NPN

NEW DATA VALUES

RES	1	5	.1000E-01	10.000	0
RES	4	2	.2000E+05	0	0
RES	2	5	.3000E+04	0	0
RES	3	5	.3000E+03	0	0
RES	1	4	.5000E+04	0	0
TE	4	2	3 .9900	.2600	.4900E-11 .1900E-09 NPN

NODE	VOLTAGE
1	10.000
2	3.333
3	0
4	4.444

BRANCH	VOLTAGE	CURRENT	POWER
1	10.000	1333.333	.1333E+05
2	1.111	.001	.001
3	3.333	.001	.004
4	0	0	0
5	6.666	.001	.006
6	-3.333	0	0
7	1.111	0	0

NODE	VOLTAGE
1	10.000
2	3.333
3	-.001
4	4.444

BRANCH	VOLTAGE	CURRENT	POWER
1	10.000	1333.333	.1333E+05
2	1.111	.001	.001
3	3.333	.001	.004
4	-.001	-.001	.000
5	6.666	.001	.006

3	.342		
4	1.022		
BRANCH	VOLTAGE	CURRENT	POWER
1	[illegible]	-359.592	[illegible]
2	.435	.111	.048
3	1.207	.000	.000
4	.322	.111	.036
5	8.377	.111	.934
6	-.809	-20.710	15.436
7	.415	18.523	7.683

P.N.P TRANSISTOR AMP. AT 300 DEG. KELVIN

CIRCUIT DATA

TYPE	NODE1	NODE2	NODE3	OHMS	VOLT	CURRENT		
RES	1	5		.100E+01	5.000	0		
RES	1	2		.300E+04	0	0		
RES	2	5		.300E+04	0	0		
RES	4	5		.250E+03	0	0		
RES	1	3		.300E+03	0	0		
TE	4	2	3	.9900	.2600	.490E-11	.190E-09	PNP
RES	1	5		.100E+01	5.000	0		
RES	1	2		.300E+04	0	0		
RES	2	5		.300E+04	0	0		
RES	4	5		.250E+03	0	0		
RES	1	3		.300E+03	0	0		
TE	4	2	3	.9900	.2600	.490E-11	.190E-09	PNP

NODE	VOLTAGE
1	5.000
2	1.898
3	3.113
4	1.573

BRANCH	VOLTAGE	CURRENT	POWER
1	5.000	999.993	4999.890
2	3.102	.001	.003
3	1.898	.001	.001
4	1.673	.007	.011
5	1.887	.006	.012
6	1.215	.005	.008
7	1.225	.008	.002

NODE	VOLTAGE
1	5.000
2	.892
3	4.285
4	.864

BRANCH	VOLTAGE	CURRENT	POWER
1	5.000	999.995	4999.944
2	4.108	.001	.016
3	.892	.000	.000
4	.864	.003	.003
5	.715	[illegible]	[illegible]
6	[illegible]	[illegible]	[illegible]

	.237	
	4.641	
	.510	

BRANCH	VOLTAGE	CURRENT	POWER
1	5.000	999.998	4999.968
2	4.763	.002	.008
3	.237	.000	.000
4	.510	.002	.001
5	.159	.001	.000
6	4.604	-.004	-.019
7	.272	.016	.004

NODE	VOLTAGE
1	5.000
2	2.145
3	2.096
4	2.479

BRANCH	VOLTAGE	CURRENT	POWER
1	5.000	999.969	4999.841
2	2.855	.001	.003
3	2.145	.001	.002
4	2.479	.010	.025
5	2.904	.010	.028
6	2.249	14.256	.695
7	.334	14.136	4.719

NODE	VOLTAGE
1	5.000
2	1.858
3	2.509
4	2.183

BRANCH	VOLTAGE	CURRENT	POWER
1	5.000	999.991	4999.860
2	3.142	.001	.003
3	1.858	.001	.001
4	2.183	.009	.019
5	2.491	.008	.021
6	.651	190.518	124.080
7	.325	-188.600	-61.213

NODE	VOLTAGE
1	5.000
2	1.858
3	2.509
4	2.183

2	3.142	.001	.003
3	1.859	.001	.001
4	1.183	.009	.019
5	2.491	.008	.021
6	.651	190.518	124.080
7	.325	-185.600	-61.213

FIG. A5

TERACTIVE COMPUTER AIDED DESIGN/RUNALA,PART,NSC THESIS 1981,AHMADU BELLO UNI

ULATOR DESIGN ANALYSIS

TYPE	CIRCUIT DATA NODE1	NODE2	OHMS	VOLT	CURRENT		
S	E	7	.2[illegible]	12.000	0		
S	1	7	.100E-01	12.000	0		
S	1	2	.2[illegible]E+04	0	0		
S	3	4	.680E+03	0	0		
S	3	5	.750E+03	0	0		
S	3	7	.100E+04	0	0		
S	5	7	.2[illegible]E+05	0	0		
D	4	7	6.800				
	6	2	3 .9900	.2500	.490E-11	.19.E-09	NPN
	2	5	4 .9900	.2500	.490E-11	.19.E-09	NPN
	NEW DATA VALUES						
S	E	7	.2[illegible]	12.000	0		
S	1	7	.2[illegible]	12.000	0		
S	1	2	.2[illegible]E+04	0	0		
S	3	4	.680E+03	0	0		
S	3	5	.750E+03	0	0		
S	3	7	.2[illegible]E+03	0	0		
S	5	7	.100E+05	0	0		
D	4	7	6.800				
	6	2	3 .9900	.2500	.490E-11	.19.E-09	NPN
	2	5	4 .9900	.2500	.490E-11	.19.E-09	NPN

NODE	VOLTAGE
1	12.000
2	8.782
3	2.599
4	4.872
5	4.564
6	12.000

BRANCH	VOLTAGE	CURRENT	POWER
1	12.[illegible]	2399.991	* .2E+05
2	12.000	2399.998	* .2E+05
3	3.218	[illegible]	.[illegible]
4	2.273	[illegible]	.[illegible]
5	1.965	[illegible]	.[illegible]
6	2.599	.813	.034
7	4.564	.000	.[illegible]
8	4.872	.012	.059

3	16.202	2.107	.043
2	3.218	.009	.033
3	.318	2.001	2.000
4	4.218	.154	.026

NODE	VOLTAGE
1	12.000
2	10.772
3	9.046
4	7.878
5	9.762
6	12.222

BRANCH	VOLTAGE	CURRENT	POWER
1	12.000	2399.999	* .2E+05
2	12.222	2399.999	* .2E+05
3	12.222	2399.999	* .2E+05
4	12.222	2399.999	* .2E+05
5	1.229	.011	.011
6	1.169	.002	.002
7	.726	.011	.014
8	7.878	.054	.422
9	1.724	.020	.035
10	1.229	.000	.000
11	.884	.001	.001
12	12.78	.22	.02

NODE	VOLTAGE
1	12.025
2	* .1E+06
3	* .3E+06
4	* .4E+06
5	* .3E+06
6	12.277

BRANCH	VOLTAGE	CURRENT	POWER
1	12.277	2427.700	* .2E+05
2	12.277	2427.700	* .2E+05
3	12.277	2427.700	* .2E+05
4	12.025	2402.925	* .2E+05
5	* .2E+06	2402.925	* .7E+07
6	* .7E+05	-2402.041	* .8E+07

7	* .8E+15	91.635	* .6E+17
8	* .4E+06	144.025	* .8E+08
9	* .2E+06	21.666	* .5E+17
10	* .1E+06	-83.102	* .1E+08
11	* .2E+06	31.985	* .4E+17
12	* .1E+06	-123.021	* .2E+08

NODE	VOLTAGE
1	12.000
2	11.811
3	11.557
4	6.350
5	11.803
6	12.000

BRANCH	VOLTAGE	CURRENT	POWER
1	12.000	2399.997	* .2E+05
2	12.000	2399.997	* .2E+05
3	12.000	2399.997	* .2E+05
4	12.000	2400.000	* .2E+05
5	.191	.000	.000
6	5.247	.048	.248
7	4.246	.044	.188
1	6.350	7346.994	* .4E+05
2	5.253	3.424	.362
3	.190	1.410	.268
4	15.453	.002	.034
5	.006	.000	.000

NODE	VOLTAGE
1	12.000
2	7.139
3	6.190
4	6.350
5	6.602
6	12.000

BRANCH	VOLTAGE	CURRENT	POWER
1	12.000	2399.999	* .2E+05
2	12.000	2399.999	* .2E+05
3	12.000	2399.999	* .2E+05

	12.000	2399.998	* .2E+05
	4.861	.002	.010
	-.166	-.000	.000
	-.412	-.001	.000
	6.350	7346.991	* .4E+05
	-.949	-5.385	5.112
	4.861	-5.333	25.923
	-.252	-1.422	.358
	.538	-1.407	.757

NODE	VOLTAGE
1	12.000
	7.835
	6.886
	6.350
	7.301
	12.000

BRANCH	VOLTAGE	CURRENT	POWER
1	12.000	2399.999	* .2E+05
	12.000	2399.999	* .2E+05
	12.000	2399.998	* .2E+05
	4.165	.002	.009
	.535	.001	.000
	-.416	-.001	.000
	6.350	7346.991	* .4E+05
	-.951	-5.385	5.115
	4.165	5.334	22.213
	-.951	-5.386	5.127
	.534	5.336	2.849

NODE	VOLTAGE
1	12.000
	7.835
	6.886
	6.350
	7.301
	12.000

BRANCH	VOLTAGE	CURRENT	POWER
1	12.000	2399.999	* .2E+05

2	12.000	2399.999	* .2E+05
3	12.000	2399.999	* .2E+05
4	12.000	2399.998	* .2E+05
5	4.165	.002	.009
6	.536	.001	.000
7	-.416	-.001	.000
8	6.350	7346.991	* .4E+05
9	6.958	25.385	5.115
10	4.165	5.334	22.213
11	-.951	-5.392	5.127
12	.534	5.336	2.849

APPENDIX IV

CONSTRUCT OF THE D.C. MODE COMPUTER AIDED DESIGN OF ELECTRONIC NETWORKS

A listing of the main and subroutine program is given in Fig. A7 and the flow-chart in Fig.A.1.

THE MAIN PROGRAM

Program lines 1 to 90 are normal house-keeping routines (see Fig.A1)

Program lines 25 and 26 define the program source code.

Program lines 28 to 31 initiates the [illegible] input.

Program lines 35 to 44 are the temperature initialisation routine.

Program lines 45 to 49 is the initialisation routine to determine whether new analysis with new sets of data is what is required or an old analysis, requiring new set of data is to be executed.

Program lines 60 to 89 initiates the variables against possible garbage.

Program lines 96 to 159 reads in the input data, the numeric portions of which are unformatted, and stores the data on tape 8.

FIG.AI

```
IVERSITY OF MINNESOTA FORTRAN COMPILER (VERSION 5.1 - 77/03/24) ON THE CYBER ORDER  0 0 1
     ONE,I=MACH,B=BOMB,T.
      0    0000000          PROGRAM CAD(INPUT,OUTPUT,DATA8,DATA1,DATA2,TAPE9=
     1    INPUT,TAPE5=OUTPUT,TAPE8=DATA8,TAPE1=DATA1,TAPE2=DATA2)
      0.    0100000B        DIMENSION TITLE(20)
      0.    0100000B        DIMENSION X(25),V2(25),V3(25),V5(25),V4(25) ,Z1(25)
      0.    0100000B        DIMENSION CP(25),V6(25),V9(25),CR(25),R0(25),Z5(25)
      0.    0100000B        DIMENSION D0(25),V7(25),D5(25),D6(25),D7(25),V8(25)
      0.    0100000B        DIMENSION V14(25),V11(25),Z3(25),V10(25),Z15(25)
      0.    0100000B        DIMENSION D8(25),Z2(25),Z4(25),Z6(25),Z16(25)
      0.    0100000B        COMMON F(25,25),C(25,25),G(25,25),B(25,1),H(25,1)
      0.    0100000B        COMMON D(25,1),A(25,1),Y(25,25),L3,M3,L
      0.    0100000B        INTEGER CG,LX,M3,X8,AS7,AS8
      0.    0100000B        EQUIVALENT CONTROL(I,J,L=0,T,U)
      0.    0100000B        EQUIVALENT F(A(1,1),V,P)
      0.    0100000B        EQUIVALENT (D(1,1),CG,DP,Z)
      0.    0100000B        REAL TT,CX,CP,R0,B6,DMP,X1,ALP,X3,Z2,Z3,Z15,KJ,DU,AR,DI
      0.    0100000B        INTEGER ANS,X8C,AS6,X111,XM12
      0.    0100000B        LITERAL AS5,X8,DP
      0.    0103100B        DATA TYPE1/"DIO"/,TYPE2/"RES"/,TYPE3/"TE"/,TYPE4/"FET"/
      0.    0103100B        DATA AS1/"REMOVE"/
      0.    0103100B        DATA AS2/"CHANGE"/
      0.    0103100B        DATA AS7/"CHOICE"/,AS8/"ERROR"/
      0.    0103100B        DATA X111/"PFET"/,X8_2/"NFET"/,AS3/"NO"/
      0.    0103100B        DATA XM12/"PNP"/,X8C/"NPN"/
      0.    0103100B        DATA ANA1/"YES"/,ANA2/"NO"/
      0.    0103100B        WRITE(6,7)
      0.    0117648   7     FORMAT(////* INTERACTIVE COMPUTER AIDED DESIGN/RUMALA,PART,MSC
     1    THESIS 1981,AHMADU BELLO UNIVERSITY*////)
      0.    0117648         WRITE(5,44)
      0.    0117708         REWIND5
      0.    0117748         READ(5,66)TITLE
      0.    0117746         WRITE(6,66)TITLE
      0.    0117748   44      FORMAT(*   DESIGN TITLE*)
      0.    0117748   66      FORMAT(20A6///)
      0.    0117748         REWIND6
      0.    0117778         WRITE(6,17)
      0.    0117458   17      FORMAT(* IS OPERATING TEMPERATURE  ROOM TEMPERATURE?*)
      0.    0117478         REWIND5
      0.    0117478         READ(5,12)CG
      0.    0117568         REWIND6
      0.    0117608         IF(CG.EQ.ANA1)GOTO17.
      0.    0117648         WRITE(6,110)
      0.    0117708   110     FORMAT(* INPUT OPERATING TEMPERATURE *)
      0.    0117718         REWIND5
      0.    0117728         READ(5,*)TT
      0.    0117768         VBOA=-.2304*TT
      0.    0120039   170     WRITE(6,303)
      0.    0120138   303     FORMAT(///*  NEW ANALYSIS?*)
      0.    0120138         REWIND5
      0.    0120158         READ(5,12)CG
      0.    0120248         IF(CG-ANA2)13,304,13
      0.    0120258   13      WRITE(6,11)
      0.    0120338         REWIND5
      0.    0120378         READ(5,*)DMP,DP,L3,LX,M3
      0.    0120428   11      FORMAT(* DAMP,FACTOR,DPT SOURCE#,TOTAL BRANCH#,TOTAL LINEAR
     1    ELEMENT#,TOTAL NODES*)
      0.    0120418         READ(5,*)DMP,DP,L3,LX,M3
      0.    0120528         WRITE(6,1)DMP,DP,L3,LX,M3
     C  L3=NO OF BRANCHES,DP=NO OF DEPENDENT SOURCES
     C  LX=NO OF LINEAR ELEMENTS,M3=NO OF NODES
      0.    0120678   304     REWIND5
      0.    0120728         READ(5)DMP,DP,L3,LX,M3
      0.    0121108         REWIND6
      0.    0121128         REWIND8
     C                 INITIALISE VARIABLES
      0.    0121148         DO1L2J=1,L3
      0.    0121238         X1(J)=0.
      0.    0121278         V1(J)=0.
      0.    0121338         V13(J)=0.
      0.    0121378         V14(J)=0.
      0.    0121438         G14(J)=0.
      0.    0121478         Z14(J)=0.
      0.    0121528         Z16(J)=0.
      0.    0121578         G15(J)=0.
      0.    0121638         H(J,1)=0.
      0.    0121678         B(J,1)=0.
      0.    0121738         VC(J)=-.6
      0.    0122018         R6(J)=0.4
      0.    0122058         G4(J)=0.
      0.    0122128         Z4(J)=0.001
      0.    0122178         G4(J)=0.01
```

```
      V9(J)=...4
      G9(J)=...0
      Z3=...021
      Z6(J)=...
  220 G6(J)=...
      AL9=1
      Z3=0.
      V9=0.
      N=...
      KJ=...
      G2=0
      U2=...
      G2=1
      U=...
      IF(CC.EQ.AMA2)GOTO400
      REWIND5
      WRITE(15,19)
      REWIND5
C     FORM INCIDENCE MATRIX
      J=0
  10  KJ=0
      READ(5,12)WHICH
      WRITE(8,12)WHICH
      J=J+1
      IF(TYPE3.EQ.WHICH)GOTO31
      IF(TYPE4.EQ.WHICH)GOTO32
      IF(TYPE1.EQ.WHICH)GOTO33
      IF(TYPE3.EQ.WHICH)GOTO34
  31  READ(5,*)G,U,X,V,Z
      WRITE(8,53)G,U,X,V,Z
      GOTO73
  32  READ(5,*)CC,BB,EE,VP,CU,ALP
      WRITE(8,57)CC,BB,EE,VP,CU,ALP
      READ(5,12)XM
      WRITE(8,12)XM
      G=EE
      U=BB
      GOTO71
  740 G=CC
      U=BB
      J=J+1
  71  KJ=KJ+1
      IF((TYPE4.EQ.WHICH).AND.(KJ.LT.2).AND.
     1 (XM2.EQ.XM))GOTO74
      IF((TYPE4.EQ.WHICH).AND.(KJ.LT.2).AND.
     1 (XM1.EQ.XM))GOTO74
      GOTO73
  33  READ(5,*)BB,EE,V
      WRITE(8,55)BB,EE,V
      GOTO73
  34  READ(5,*)CC,BB,EE,ALP,AP,CU,KU
      WRITE(8,56)CC,BB,EE,ALP,AP,CU,KU
      READ(5,12) XM
      WRITE(8,12)XM
      G=EE
      U=BB
      GOTO71
  74  G=CC
      U=63
      J=J+1
  71  KJ=KJ+1
      IF(TYPE3.EQ.WHICH.AND.(KJ.LT.2).AND.XM2.EQ.XM)GOTO74
      IF(TYPE3.EQ.WHICH.AND.(KJ.LT.2).AND.XM1.EQ.XM)GOTO74
  73  IF(J-L3)10,204,204
 204  ENDFILE8
C     SET UP PRIMITIVE ADMT. MTX FOR THE LIN. BRANCHES
 400  X3=0.
      M5=7
      REWIND1
      READ(1) DMP,DP,L3,LX,M3
      DO 27 I=1,L3
      DO 27 J=1,L3
 27   A(I,J)=0.
      IF(M.GE.1)GOTO401
      WRITE(5,19)
 19   FORMAT(*          CIRCUIT   DATA *)
      WRITE(5,427)
 401  J=0
      REWIND2
      REWIND8
      IF(AA1.EQ.CC)GOTO98
      REWIND1
      READ(1)DMP,DP,L3,LX,M3
```

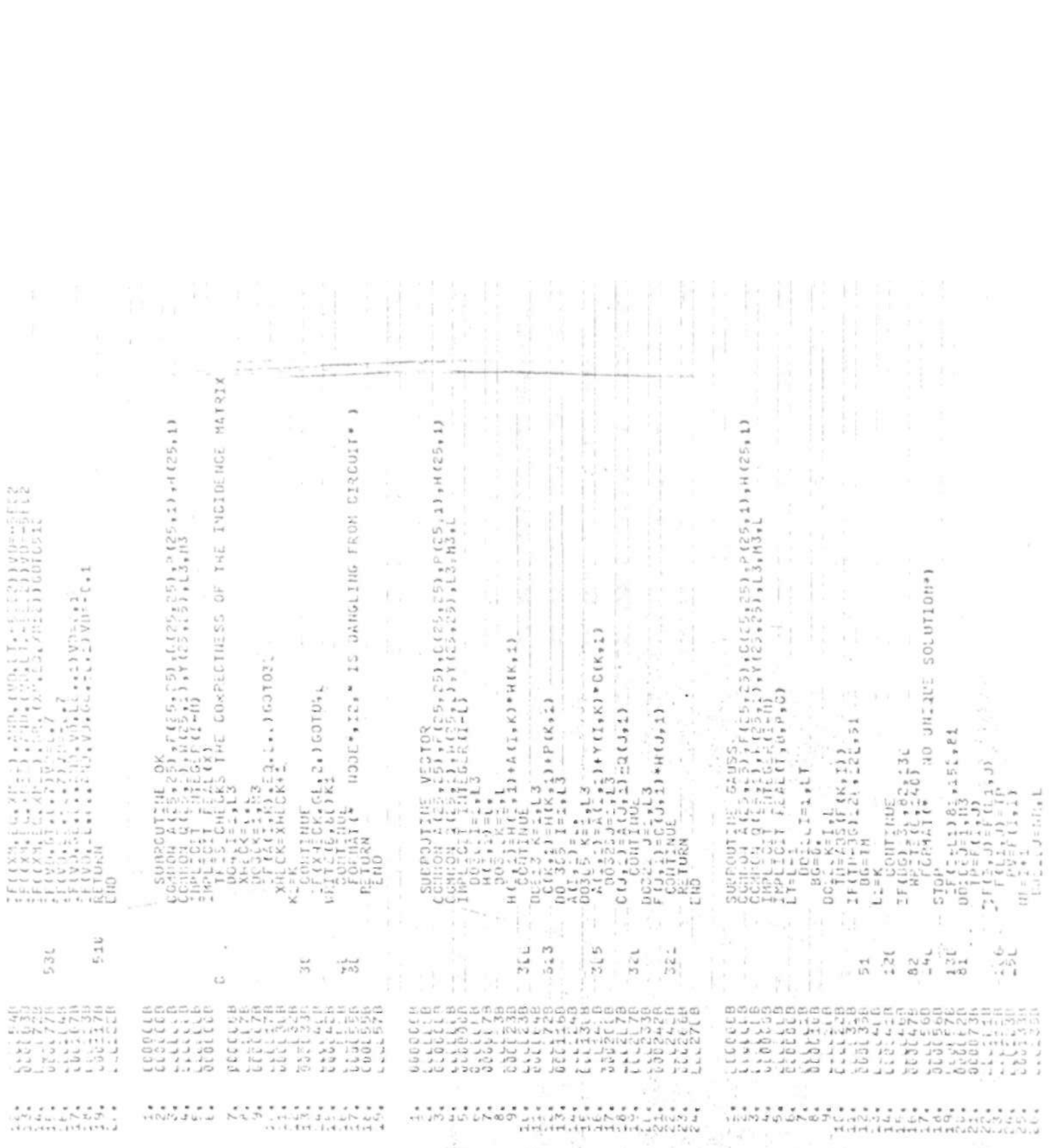
```
157.    0127478   98      J=J+1
158.    0127538           KJ=J
159.    0127638           V=*
160.    0127738           Z=*
161.    0127838           IF(Z.GT.2)WHICH
162.    0127948           IF(TYPE1.EQ.WHICH)GOTO123
163.    0128048           IF(TYPE2.EQ.WHICH)GOTO124
164.    0128158           IF(TYPE3.EQ.WHICH)GOTO125
165.    0128258           IF(TYPE4.EQ.WHICH)GOTO126
166.    0128368   123   READ(8,55)BB,EE,V
167.    0128468           G=BB
168.    0128568           U=EE
169.    0128668           IF(ANS.EQ.AS1)GOTO216
170.    0128768           IF((ANS.EQ.AS2).AND.(M.GT.1))GOTO127
171.    0128868           IF((ANS.EQ.AS8).AND.(M.GT.1))GOTO127
172.    0128968           IF((B3.EQ.G2).AND.(EE.EQ.G2).OR.(EE.EQ.G2)
     1  .AND.(BB.EQ.U.))GOTO216
173.    0129068           IF((ANS.EQ.AS7).OR.(ANS.EQ.AS8))GOTO999
174.    0129168           GOTO 216
175.    0129268   122     IF((BB.EQ.G2).AND.(EE.EQ.U2).OR.((EE.EQ.G2).AND.(BB.EQ.U2)))G1
176.    0129368           IF((BB.EQ.G2).AND.(EE.EQ.U2).OR.((EE.EQ.G2).AND.(BB.EQ.U2)))U1
177.    0129468           IF(ANS.EQ.AS8)GOTO 8
178.    0129568
179.    0129668           G=G1
180.    0129768           U=U1
181.    0129868           V=X2
182.    0129968   999     WRITE(2,55)G,U,V
183.    0130068           GOTO205
184.    0130168   127     PRINT
185.    0130268           IF(M6.GT.2)GOTO777
186.    0130368           REWIND2
187.    0130468           ME=ME+1
188.    0130568   777     READ(2,55)G,U,V
189.    0130668   216     A(J,G)=1
190.    0130768           A(J,U)=-1
191.    0130868           IF(M.GE.1)GOTO8
192.    0130968           WRITE(6,374)WHICH,G,U,V
193.    0131068           GOTO6
194.    0131168   124     READ(8,56)CC,BB,EE,ALP,AR,CU,KU
195.    0131268           READ(8,12)XM
196.    0131368           IF(M.GE.1)GOTO9
197.    0131468           WRITE(6,372)WHICH,CC,BB,EE,ALP,AR,CU,KU,XM
198.    0131568   9       G=EE
199.    0131668           U=BB
200.    0131768           GOTO95
201.    0131868   92      G=CC
202.    0131968           U=BB
203.    0132068           J=J+1
204.    0132168   96      A(J,G)=1
205.    0132268           A(J,U)=-1
206.    0132368           KJ=KJ+1
207.    0132468           IF(TYPE2.EQ.WHICH.AND.(KJ.LT.2).AND.XM2.EQ.XM)GOTO92
208.    0132568           IF(TYPE3.EQ.WHICH.AND.(KJ.LT.3).AND.XM1.EQ.XM)GOTO92
209.    0132668           GOTO6
210.    0132768   125     READ(8,53)G,U,X,V,Z
211.    0132868           IF(ANS.EQ.AS2)GOTO215
212.    0132968           IF((ANS.EQ.AS2).AND.(M.GT.2))GOTO224
213.    0133068           IF((ANS.EQ.AS2).AND.(M.GT.1))GOTO224
214.    0133168           IF((G.EQ.G2).AND.(U.EQ.U2).OR.((G.EQ.U2).AND.(U.EQ.G2)))
     1  GOTO218
215.    0133268           IF((G.EQ.G2).AND.(U.EQ.U2).OR.((G.EQ.U2).AND.(U.EQ.G2)))G1=G2
216.    0133368           IF((G.EQ.G2).AND.(U.EQ.U2).OR.((G.EQ.U2).AND.(U.EQ.G2)))U1=U2
217.    0133468           IF(ANS.EQ.AS1)GOTO 8
218.    0133568           G=G1
219.    0133668           U=U1
220.    0133768           X=V3
221.    0133868           V=X3
222.    0133968   888     WRITE(2,53)G,U,X,V,Z
223.    0134068           GOTO215
224.    0134168   224     PRINT
225.    0134268           IF(M6.GT.2)GOTO666
226.    0134368           REWIND2
227.    0134468           ME=ME+1
228.    0134568   666     READ(2,53)G,U,X,V,Z
229.    0134668   215     Q(J,1)=Z
230.    0134768           Y(J,J)=1/X
231.    0134868           Y(J,1)=V
232.    0134968           A(J,G)=1
233.    0135068           A(J,U)=-1
234.    0135168           IF(M.GE.2)GOTO3
```

```
241.    .13955B           WRITE(6,372)WHICH,G,U,X,V,Z
242.    .1395?B            GOTO3
243.    .1395?B   226       READ(8,57)CC,BB,EE,VP,CU,ALP
244.    .1396?B            BB=BB(8,12)XB
245.    .1396?B            IF(U.GE.1)GOTO3C
246.    .1396?B            WRITE(6,375)WHICH,CC,BB,EE,VP,CU,ALP,XM
247.    .1396?B   9.C      G=GL
248.    .1396?B            U=BB
249.    .1396?B            GOTO963
250.    .1396?B   92L       G=CC
251.    .1396?B            U=BB
252.    .1396?B            J=J+1
253.    .1396?B   960       A(J,G)=1
254.    .1396?B            A(J,U)=-1
255.    .1396?B            KJ=KJ+1
256.    .1396?B            IF((TYPE4.EQ.WHICH).AND.(KJ.LT.2).AND.
                1          (XM22.EQ.XB))GOTO92L
257.    .1397?B            IF((TYPE4.EQ.WHICH).AND.(KJ.LT.3).AND.
                1          (XM22.EQ.XB))GOTO92L
258.    .1397?B   8        IF(J-L)98,2CC,2CC
259.    .1397?B   2C2      DB=DB+1
260.    .1397?B            IF(U.GE.1.OR.MS.GT.1)GOTO2L3
261.    .1397?B            WRITE(6,2C3)
262.    .1397?B            CHOICE=1
263.    .1397?B            RETURN3
264.    .1397?B   2C1      FORMAT(' WANT TO CHANGE VALUE OF OR REMOVE A BRANCH?')
265.    .1397?B            READ(5,12)ANS
266.    .1397?B
267.    .1397?B
268.    .1397?B
269.    .1397?B            IF(ANS.EQ.ANS)GOTO 401
270.    .1397?B            WRITE(6,6L7)
271.    .1397?B            RETURN3
272.    .1397?B   507      FORMAT('                NEW DATA VALUES ')
273.    .1397?B   5,7      FORMAT(' NODE1 NODE2 VOLTAGE  IMPEDANCE')
274.    .1397?B            READ(5,*)G2,U2,X3,V3
275.    .1401?B            WRITE(6,5,7)
276.    .1401?B            RETURN3
277.    .1401?B            GOTO4,1
278.    .1401?B   283      IF(DP.EQ.1)GOTO22
279.    .1401?B            CALL SIN(BB,CC,ANA2,DP)
280.    .1402?B
C                          SET UP PRIM. ADMT. MATRIX FOR NONLIN. BRANCHES
281.    .1402?B   22       IF(LX.EQ.L3)GOTO64
282.    .1402?B
283.    .1402?B            J=6
283.    .1402?B            REWIND8
284.    .1402?B   45    READ(8,12)WHICH
285.    .1403?B            KJ=L
286.    .1403?B            V=C.
287.    .1403?B            Z=C.
288.    .1404?B
289.    .1404?B            J=J+1
290.    .1404?B            IF(TYPEC.EQ.WHICH)READ(8,57)G,U,X,V,Z
291.    .1404?B            IF(TYPE4.EQ.WHICH)READ(8,57)CC,BB,EE,VP,CU,ALP
292.    .1404?B            IF(WHICH.EQ.TYPE3)GOTO63
293.    .1404?B            IF(U.EQ.1)VD=CC*4
294.    .1404?B            IF(U.EQ.1)DI=-.12
295.    .1404?B            IF(U.EQ.1)DP=C.12
296.    .1404?B            IF(WHICH.EQ.TYPE4)GOTO632
297.    .1404?B            IF(TYPE1.EQ.WHICH)GOTO58(8
298.    .1404?B            READ(8,56) CC,BB,EE,ALP,AP,CU,KU
299.    .1404?B            READ(8,12)XP
300.    .1404?B            IF(TYPE3.EQ.WHICH)GOTO52
301.    .1404?B   5,8        IF(TYPE.EQ.WHICH)READ(8,55)BB,EE,V
C                          DAMP THE NEWTON ITERATION
302.    .1404?B            IF(DP.GT.1.)VD=H(J,1)
303.    .1404?B            CU=1E-39
304.    .1404?B            VD(J)=V0(J)
305.    .1404?B            CALL LIMIT(VD,J,N,V2,XM,XM11,XM12,XM1,XM2)
306.    .1404?B            V8(J)=VD
307.    .1404?B            VD=V0(J)
308.    .1404?B            DI=CU*(EXP(VMDA*VD)-1)
309.    .1404?B            GD=VMDA*DMP*(DI+CU)
310.    .1404?B            IF(VD.GT.L.)GD=VMDA*DMP*CU*EXP(VMDA*VD)
311.    .1404?B           -GA(J)=DI
312.    .1404?B           -GD=GA(J)
313.    .1404?B            Y(J,J)=GD*DMP
314.    .1404?B            Z=DMP*(GD*VD-DI)
315.    .1404?B            G(J)=GA(J)
316.    .1404?B             V(J)=V8(J)
317.    .1404?B            GOTO
318.    .1404?B   3L        IF((N.GT.1).AND.(XM.EQ.XM12))VD=-H(J,1)
319.    .1404?B            IF((N.GT.1).AND.(XM.EQ.XM12))VD=H(J,1)
```

```
      V6(J)=VE(J)
      CALL LIMIT(VD,J,H,V6,XH12,XH2,XH1,XH2)
      V7(J)=VD
      VD=V6(J)
      DA=(H(J,1)*(XP(VHDA*VD1-1)
      GD=VHDA*DHP*(D1-GD)
      IF(VD).GT.0.)GD=VHDA*
      G7(J)=GD
      G6=GD(J)
      L5=J+2
      Y(L5,J)=-ALP*DHP*GD
      Y(J,J)=GD+DHP
      Z=GHP*VD+D1-GD
      Z=Z*(1.0+ZK(J)*Z+D1-GD
      F(X(J)=Y7(J)
      Z=Z(J)
      Z=DHP*(Z+AK*Z3)
      Z6(J)=Z2
      G6(J)=G7(J)
      GO TO 37
   37 J=J+2
      L4=J-2
      IF((X(J).EQ.XH(J).AND.(X(J).EQ.XH2).AND.(H(J,1).EQ.0.))VD=VD1
      IF((X(J).EQ.XH(J).AND.(X(J).EQ.XH2).AND.(H(J,2).EQ.0.))VD=VD
      V2(J)=V4(J)
      CALL LIMIT(VD,J,H,V6,XH2,XH1,XH2)
      V5(J)=VD
      VD=V6(J)
      D1=GD*(EXP(VHDA*VD1-1.)
      GD=VHDA*DHP*(D1-GD)
      IF(VD).GT.0.)GD=VHDA*D1
      G5(J)=GD
      GD=G4(J)
      Y(J,J)=GD*DHP
      Z=GD*VD+DHP*GD
      F(X(J)=EQ.XH(J)Z=D1-GD*
      Z2=Z
      V4(J)=V5(J)
      Z=DHP*(Z-ALP*Z1D)
      Z6(J)=Z3
      Z6(J)=G5(J)
      GO TO 630
      B=A(13,12)XM
      IF((H(J,1).AND.(X(J).EQ.XH2).AND.(H(J,2).EQ.0.))VD=H(J,1)
      IF((H(J,1).AND.(X(J).EQ.XH2).AND.(H(J,2).EQ.0.))VD=H(J,1)
      V2(J)=V6(J)
      CALL LIMIT(VD,J,H,V2,XH1,XH12,XH2)
      VD=V2(J)
      VD=VD
      IF(X(J).EQ.XH(J))VD=VD-VP
      IF(VD.GT.VP)VD=VD-VP
      IF(VD.LT.VP)D1=-GD*(VD-VP)
      GD=-D1*(VD-VP)
      GD=GD
      G7(J)=GD
      GD=Z*Y(J)
      Y6=GD*(VD**2-VP**2)
      Z=GD*(VD**2-VP**2)
      Z2=Z
      Z4=DHP*(Z-Z13)
      Z=Z-Z6(J)
      Z=DHP*(Z-Z13)
      Z=Z
      G6(J)=Z6(J)
      GO TO 470
      J=J+2
      L4=J-2
      IF((H(J,1).AND.(X(J).EQ.XH2).AND.(H(J,2).EQ.0.))VD=H(J,1)
      IF((H(J,1).AND.(X(J).EQ.XH2).AND.(H(J,2).EQ.0.))VD=H(J,1)
      V2(J)=V6(J)
      CALL LIMIT(VD,J,H,V12,XH1,XH12,XH2)
      VD=V2(J)
      VD=VD
      IF(X(J).EQ.XH(J))VD=VD-VP
      VDS=SD*(VD-VP)**2
      IF(VD.GT.VP)D1=-GD*(VD-VP)
      IF(VD.LT.VP)D1=-C
      GD=2*GD*(VD-VP)+VDS*D1=-C
      G6(J)=GD
```

```
      GD=G24(J)
      Y(J,J)=GD*DMP
      Y(L24,J)=-HALF*DMP*GD
      Z=-DMP*(VD**2-V**2)
      Z23=Z
      V24(J)=V15(J)
      Z=Z24(J)
      Z=DMP*(Z-Z2L)
      Z24(J)=Z23
      G24(J)=G15(J)
C     FORM THE PRIMITIVE NODAL CURRENT AND VOLT. VECTOR
   31   P(J,1)=V
        Q(J,1)=Z
      V=0.
      KJ=KJ+1
      IF(KJ.GT.1)GOTO63
      IF(DHECH.EQ.TYPE1)GOTO63
      IF(XM.EQ.XM)GOTO37
      IF(XM1.EQ.XM)GOTO37
      IF(XM.EQ.XM11)GOTO47
      IF(XM.EQ.XM12)GOTO47
   63 IF(J-L3)45,64,64
   64 CALL OK
      L=M2+1
      CALL CUVO
      CALL GAUSS
      CALL VECTOR
C       V(J)=X(J) = INCREMENT IN JTH NODE VOLTAGE
C     TEST CONVERGENCE CRITERION
      ERR=0.0
      IF(LX.EQ.L3)GOTO412
      M=M+1
      IF(M.LT.4)GOTO412
      KJ=0
        DO6J=1,M3
      X1(J)=X(J,1)
      ERR=ERR+(V2(J)-X1(J))**2
    5 IF(ERR.LT.1.E-10)KJ=KJ+1
      DO1E J=1,M3
    16 V1(J)=X1(J)
  412 WRITE(6,24C)
      DO32 J1=1,L
      WRITE(6,360)J1,N(J1,1)
  33C   CONTINUE
        WRITE(6,37L)
        DO385J2=1,L3
      WRITE(6,390)J2,H(J2,1),G(J2,1),F(J2,1)
  385   CONTINUE
      LE=NZ+2
      IF((KJ.GE.LE).AND.(M.GT.3))GOTO611
      IF(M.GT.3LL)GOTO611
      IF(LX.NE.L3)GOTO412
  417   FORMAT(* TYPE*,2X,*NODE1*,1X,*NODE2*,5X,*OHMNS*,7X,*VOLT*,
     1  7X,*CURRENT*)
   56 FORMAT(I4,2X,I4,2X,I4,2X,F5.4,2X,F5.4,
     1  2X,F8.3,2X,E9.3/)
   53   FORMAT(I4,2X,I4,2X,E9.3,2X,F8.3,2X,F8.3/)
   55   FORMAT(I4,2X,I4,2X,F8.3/)
   57   FORMAT(I4,2X,I4,2X,I4,2X,F8.3,2X,E9.3,2X,F8.3/)
   52   FORMAT(A6)
  54C   FORMAT(*      NODE          VOLTAGE*)
  36C   FORMAT(29,7X,F8.3/)
  371   FORMAT(2X,A4,2X,I4,2X,I4,2X,2X,E9.3,2X,F8.3,2X,F8.3/)
  372   FORMAT(1X,A4,2X,I4,2X,I4,2X,,4,2X,F5.4,2X,
     1 F5.4,2X,E9.3,2X,E9.3,2X,A6/)
  374   FORMAT(2X,A4,2X,I4,2X,I4,2X,F8.3/)
  375   FORMAT(2X,A6,2X,I4,2X,I4,2X,I4,2X,F8.3,
     1 2X,E9.3,2X,F8.3,A6/)
  39L   FORMAT(4X,12,9X,F8.3,5X,F8.3,6X,F8.3/)
  371     FORMAT(*  BRANCH       VOLTAGE       CURRENT      POWER*)
  622   CALL EXIT
        END

      SUBROUTINE LIMIT(VD,J,M,V2,XM,XM11,XM12,XM1,XM2)
      DIMENSION V2(25)
      INTEGER XM,XM11,XM12,XM1,XM2
      IF(M.EQ.0)V2(J)=VD
      VO=V2(J)
      IF(VD.GT.0.)VD=VO
      IF((VD-VO).LT.-0.2)VD=VO-0.2
      IF((VD-VO).GT.0.2)VD=VO+0.2
      IF(XM.EQ.XM1)VD=0.6
      IF(XM.EQ.XM11.AND.VD.GT.0.6)VD=0.6
      IF(XM.EQ.XM12).AND.(VD.GT.0.6))VD=0.6
```



```
      CN=F(J,I)/PV
      DO 110 K=1,NS
  110   F(J,K)=F(J,K)-CN*F(I,K)
      DO 6 I=1,L
      IV=L+I-1
      B=F(IV,NS)
      IF((IV-L),EQ,1)GOTO180
      DO 2 J=2,I
      K=NS+1-J
    2 B=B-F(IV,K)*W(K,1)
  180 W(IV,1)=B/F(IV,IV)
    6   RETURN
        END

      SUBROUTINE CURR
      COMMON A(25,25),F(25,25),C(25,25),P(25,1),H(25,1)
      COMMON Q(25,1),D(25,1),Y(25,25),L3,NS,L
      DIMENSION AT(40,40)
      IMPLICIT INTEGER(I-N)
      IMPLICIT REAL(A)
C     THIS GENERATES THE CURRENT VECTOR AND ADMITTANCE MATRIX
      N=L+1
      DO 112 I=1,L
      DO 112 J=1,L3
  112 AT(I,J)=A(J,I)
      DO 9 I=1,L
      DO 95 J=1,L3
      C(I,J)=0.
      DO 95 K=1,L3
   95 C(I,J)=C(I,J)+AT(I,K)*Y(K,J)
      DO 225 I=1,L
      DO 225 J=1,L
      F(I,J)=0.
      DO 225 K=1,L3
  225 F(I,J)=F(I,J)+C(I,K)*A(K,J)
      DO 31 I=1,L
      H(I,1)=0.
      DO 315 K=1,L3
  315 H(I,1)=H(I,1)+C(I,K)*P(K,1)
      DO 335 I=1,L
      Q(I,1)=0.
      DO 335 K=1,L3
  335 Q(I,1)=Q(I,1)+AT(I,K)*Q(K,1)
      DO 345 J=1,L
  345 H(J,1)=H(J,1)-Q(J,1)
      RETURN
      END

      SUBROUTINE GEN(N,CG,ANA2,DP)
      COMMON A(25,25),F(25,25),C(25,25),P(25,1),H(25,1)
      COMMON D(25,1),Q(25,1),Y(25,25)
      INTEGER DP,T3,F3,CG,ANA2
C     T3=TO BRANCH,F3=FRO BRANCH,V3=TRANSFER RATIO
      REWIND 8
      IF(N.GE.1)GOTO22
      WRITE(5,24)
      WRITE(6,24)
   24 FORMAT(' LINEAR DEPENDENT SOURCES"TO BRANCH,FRO BRANCH,
     1 TRANSFER RATIO')
      REWIND8
   22   DO 23 I=1,DP
      IF(CG.EQ.ANA2)GOTO21
      IF(N.GE.1)GOTO21
      READ(5,*)T3,F3,V3
      WRITE(6,23)T3,F3,V3
      WRITE(6)23)T3,F5,V3
      IF(N.EQ.1)GOTO36
      READ(8,23)T3,F3,V3
   36 Y(T3,F3)=V3
   21 CONTINUE
      ENDFILE 8
   23 FORMAT(/1X,I4,2X,I4,2X,E9.3/)
      RETURN
      END
```

Program lines 161 to 254 reads input data stored in tape8 and generates the incidence matrix.

In the advent of a command to change a diode parameter, lines 170 to 181 makes the necessary changes.

If a change of resistor parameters is noted, the necessary changes are taken care of in lines 211 to 222.

Program lines 261 to 275 initialis e the element parameter change r utines.

Program lines 281 to 421 contains the routine for generating the iterative primitive admittance matrix for diodes, bipolar transistors and FeTs; as well as the associated nodal voltages and current vectors.

Program lines 297 to 315 takes care of the diode case.

Program lines 316 to 364 takes care of the bipolar transistor case, while program lines 363 to 423 takes care of the MOSFET case.

THE SUBROUTINES

(a) Subroutine limit, bounds the possible forward diode voltages at between(0.0 to 0.7)volts magnitude. This enhances convergence, as outlined in section 2.2.
(b) Subroutine OK tests for isolated nodes and gives appropriate warning.

(c) Subroutine Gauss is a Gaussian elimination routine for solving the ensuing linearised non-linear simultaneous equations.

(d) Subroutine GEN is a routine for generating the primitive admittance matrix for linear dependent sources.

(e) Subroutine VECFOR sets up the matrix equations for use by subroutine GAUSS for evaluating the nodal voltages.

(f) Subroutine CUVO generates the branch voltage and currents, as well as the power dissipation per branch.

ASSIGNMENT OF VARIABLES:

A - incidence matrix array

P- voltage source vector

Q - current source vector

X1 - branch voltage vector initialised for convergence criterion

V1 - branch voltage vector: present value compared with X1 to test for convergence.

Y - primitive nodal admittance matrix array

F - array representing nodal admittance matrix in CUVO. In the main program, it represents power dissipation in a branch.

H - branch voltage vector

C - branch current vector -(main program only)

V - nodal voltage vector

NB - total number of branches; transistors are
counted as having two branches

ND - total number of circuit nodes

DF - damping factor

TT - operating temperature; default zero, is
room temperature.

DE - number of dependent sources

BB - base node(gate node in case of FETs)

EE - emitter node(source node in case of FETs)

CC - collector node(drain node in case of FETs)

ALF - static short circuit forward common base
current gain

IES - static saturation current with B-E junction
shorted.

10. Kuh,E... and Hajj Ibrahim N.(1971):"Non-linear Circuit Theory: Resistive Networks",Proc. IEEE Vol.59,no.3,pp340-354

11. Kuo,F.F.(1966):"Network analysis by digital Computer ",Proc. of IEEE,Vol.54,pp.820-829

12. Kuo,F.F. and Magnuson,W.G.:"Computer oriented Circuit Design",pp.31-68,pp.123-187, Prentice Hall,Inc.

13. McCalla,W.J.andPederson,D.O.(1971):"Elements of Computer Aided Circuit Analysis",Trans. IEEE,Vol.CT-18,No.1,pp14-26.

14. Ohtsuki,T.and Ishida N.(1971):"D.C. Analysis of Non-linear Network Based on Generalised Piece-wise Linear Characterisation", Trans. IEEE,Vol.CT-18,no.1

15. Rumala,S.N.(1981):"Software 'wait' enhances hardware implementation",Electronic Engineering, pp.25,Morgan-Grampian Ltd.

16. Schiman,...(1970):"Integration system of a non-linear analysis program",Trans.IEEE, vol.CT-17,no.3,pp.378-385.

17. Shier,R.D.(1981):"A computational study of Floyd's algorithm",An International journal of Computers and Operations Research", vol.18,no.4,pp275-293.

18. Stark,A.E.(1979):"Introduction to numerical methods",pp.123-190,Macmillan Publishing company,Inc.

19. Stern,T.E.(1960):"On the equations of non-linear networks",IEEE Trans. on circuit theory", vol.CT-13,PP.7481.

20. Wolfendale,E.(1970):"Computer aided design techniques",PP.51-79,Butterworth and co.