Computer aided analysis of electric circuits using state-variable techniques.

ABSON, D.S.

1976

SCHOOL OF ELECTRONIC AND ELECTRICAL ENGINEERING

ROBERT GORDON'S INSTITUTE OF TECHNOLOGY, ABERDEEN

COMPUTER AIDED ANALYSIS OF ELECTRIC CIRCUITS USING STATE-VARIABLE TECHNIQUES

Thesis submitted in the application of the degree of Master of Philosophy at Robert Gordon's Institute of Technology, awarded by the Council for National Academic Awards.

Derek S. Abson, BSc

August, 1976

DECLARATION

I hereby declare that this thesis, composed by myself, is a record of the work carried out by myself, has not been accepted in any previous application for a degree, and that all sources of information have been acknowledged.

Derek S. Abson

ACKNOWLEDGEMENTS

I would like to express my gratitude to my supervisors Mr. F.G. McIntosh, Mr. E. Forrest and Mr. J. Eades for their advice and encouragement throughout this thesis.

I would also like to acknowledge the Scottish Education Department whose Scholarship made this study possible.

CONTENTS

PAGE

DECLARATION

ACKNOWLEDGEMENTS

1. Introduction to Computer Aided Circuit Analysis 1

1.1 Role of Computer Aided Design 1

1.2 Review of some existing programs 2

1.2.1 ECAP 2

1.2.2 NET1 3

1.3 State Variable Analysis of Networks 4

1.4 Essential Features of a Program based on State Variable Techniques 5

1.5 Conclusions 6

2. State Equation Formulation 7

2.1 Introduction 7

2.2 Formulation of a Degenerate System of Equations 12

2.3 Constraints 23

2.4 Conclusions 31

3. Solution of the State Equations 33

3.1 Introduction 33

3.2 Certaine's Method 34

3.3 Gear's Method 36

3.4 Conclusion 40

4. Development Programs 41

4.1 Introduction 41

4.2 Programs FOFSVM1,2 42

4.3 Programs FOFSVM3,4 44

4.4 Program FOFSVM5 46

4.5 Program FOFSVM6 48

4.6 Conclusions 49

5. Program Tests 50

5.1 Input of Data to FOFSVM5 51

5.2 Output of Solution 56

5.3 Data Pack 57

5.4 Test Examples 58

5.5 Operating Instructions 75

5.6 Operator Procedure 77

6. Final Conclusions and Future Developments 78

Appendix

References

1. Introduction to Computer Aided Circuit Analysis

1.1 Role of Computer Aided Design

Computer programs cannot replace the circuit designer or reduce circuit design into a rote process: they are simply thereto facilitate the investigation of possible avenues of research which would otherwise take years rather than months or weeks. The basic design is still in the hands of the designer who may with the aid of the computer decide whether to accept, modify or reject his initial intuitive feelings for the problem. When the basic circuitry fits the requirements it can be optimised with respect to its basic parameters. Reliability and sensitivity analyses can also be made with the aid of the computer. The circuit can now be built and tested to see if experiment matches theory. This procedure can be adopted in various stages in the building of a complete device or unit. The designer has to interpret the information supplied by the computer and decide if the design is acceptable in its present form. The computer can be programmed to make some decisions but the major ones must be those of the designer.

1.2 Review of some existing programs

1.2.1. ECAP[1]

ECAP is a piecewise linear network analysis program consisting of three packages: DC, AC and Transient.

The DC Analysis Program performs a linear steady state nodal analysis to produce a nominal solution. This can be followed by a modified solution varying up to fifty of the circuit resistive parameters. Additional capabilities are a Partial Derivative and Sensitivity Calculation, Worse-Case Analysis for Node Voltages, Standard Deviation Calculation for Node Voltages and finally Error Checking on Residual Currents.

The AC Analysis Program performs a linear analysis of the steady state due to sinusoidal fixed frequency excitation, producing a nominal output followed by a variation of parameters resulting in a modified solution as in the D.C. Analysis Section.

The Transient Analysis Program produces a piecewise-linear representation of the original set of integro-differential equations in the form of difference equations which are solved by an implicit numerical integration technique.

ECAP is a tool to be used by the circuit designer, analyst, reliability engineer and others. As with any other tool, it must be used with engineering judgement.

1.2.2 NET 1

NET 1 is a non-linear network analysis program. It does a topological formulation of the state equations of the network to be investigated. This is preceded by a DC analysis if initial conditions are to be calculated. The solution of the set of differential equations known as the State Equations is accomplished either using Adam's method or Certaine's method. The first is a predictor corrector method of numerical integration which requires back values at three previous time intervals. Additional numerical integration procedures are used by the program to produce these back values. The second method is one based on the exponential solution of a single linear, first order differential equation, or a set of uncoupled linear first order differential equations. There is no predictor corrector mechanism in this method, but it has a variable step size routine. The choice between the two methods is made by the program. The transient integration produces only state variables directly. A subsidiary calculation produces node voltages and inductor and device currents as required by the user. Simple sets of equations would be solved by Certaine's method and complex ones by Adam's method.

1.3 State Variable Analysis of Networks[2]

Differential equations may be written in the time domain for a particular network, the resulting equations being, in general, of different orders, but by the introduction of intermediate variables they may be reduced to a set of first order differential equations. These equations are called the State Equations of the network and the independent variables known as the State Variables. The set of state variables usually chosen are a set of independent capacitor currents and inductor voltages. The state variables of any network are the minimum set of variables, which together with all inputs, are sufficient to determine all other quantities in that network at any point in time.

If the resulting set of state equations is linear with respect to the state variables, time invariant, and contains sources which are piecewise linear functions of time or constants then they can be solved using the method of Transition Matrices known as Certaine's method. Otherwise a powerful general method is required to deal with problems arising from widely separated time constants, non-linearities and error propagation due to the method of numerical integration being used.

1.4 Essential Features of a Program Based on State Variable Techniques[1]

There are three main areas in the formulation and solution of the State Equations for a general RCL network:-

(i) Topological formulation of the relevant Kirchhoff-Current-Law (KCL) and Kirchhoff-Voltage-Law (KVL) equations which comprise the fundamental cutset and tieset matrices to form an initial set of equations,

(ii) Reduction of this set of equations until the state variable formulation is achieved,

(iii) Solution of the state equations by a powerful general numerical method of integration such as that of Gear.[3]

The first section requires procedures to perform the tasks of computing the tieset and cutset matrices along with their associated trees. Also, subroutines are required to deal with the different types of source inputs and set up the initial system of equations from which the state equations are to be derived. This is the simplest of the three sections.

The second section requires routines to eliminate any unwanted variables and equations from the system already formulated. This entails counting and formulating any equations of constraint, and solving for the dependent variables by Gaussian full pivotting techniques. The remaining equations are brought into State Variable form. This section rivals the third in its complexity.

The third section seems at first hand to be the easiest, but having a set of independent differential equations to operate on it is found that different problems yield widely different complexity in solution, and in general require a powerful numerical technique. The method of solution employed by such a program needs routines which control step size, stability, error propagation and rate of convergence to the desired solution.

1.5 Conclusions

Once a network is drawn the task of producing a set of state variable equations which totally describe the network relative to some initial state is straight forward in outline, but the details became exceedingly complex. The solution of a random set of state equations needs a powerful method of numerical integration, as the simplest of problems can turn out to have some undesirable though unavoidable characteristics, which require careful checking throughout the solution process to make sure that the optimum step size within the stability interval for the numerical method chosen in being used, that the error propagation is tolerable and that the solution time is economical.[4]

It must be stressed that the program is only on aid within the design process, thus its effectiveness depends on the direct circuit knowledge and interpretation of the user.

2. State Equation Formulation

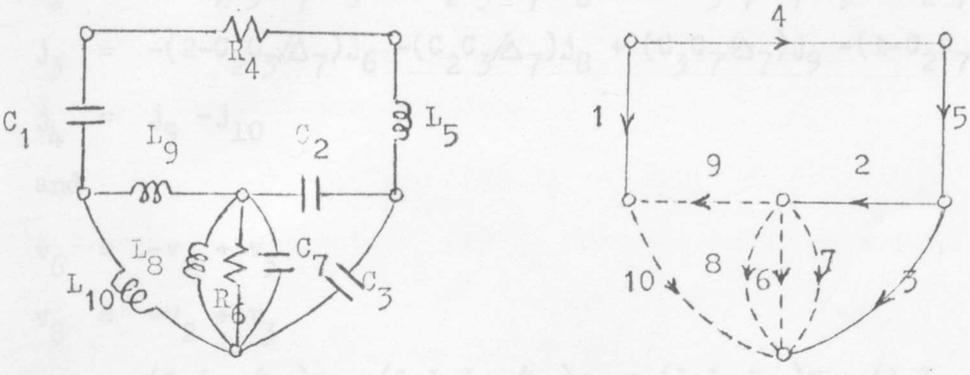
2.1 Introduction

There are numerous methods of formulating the equations which describe the behaviour of a general RCL network. One method, Bashkow's Topological Formulation leads directly to a set of first order differential equations with respect to time called the state equations. A complete set of variables plus all inputs is sufficient to specify the state of the network at any future point in time. Other methods of formulating the state equations e.g. Capacitor-node inductor-loop methods, or Supposition method do not lead directly to a first order set of differential equations.[5]

Briefly, Bashkow's Formulation consists of choosing a tree which contains the maximum number of capacitor-element branches and the minimum number of branches containing inductors, each branch containing one and only one capacitor or one inductor. Then a set of Kirchhoff current equations (one for each tree branch) and a set of Kirchhoff voltage equations (one for each tree link) are written. The relations between capacitor currents and voltages, inductor voltages and currents are substituted as well as the algebraic relations between resistive currents and voltages to produce a mixed set of algebraic, differential and integral equations.

Any loops of capacitors in the network appear as first order integral equations. Any algebraic relations involving only inductor currents also appear as first order integral equations. Loops of inductors appear as additional first order differential equations as do any algebraic relations between capacitor currents. The integral equations must be eliminated and are easily dealt with. The excess differential equations need not be eliminated unless the calculation of an inverse is required. They can in general be taken care of by Gaussian Elimination (though this is not straightforward). The resistor equations are eliminated leaving a set of differential equations in state variable form.

EXAMPLE



RCL Network and a capacitor based tree

o———————o tree branch

o- - - - - - -o tree link

$$\text{KCL}\quad\begin{aligned}
j_1 &= -j_9 + j_{10} &&= C_1 dv_1/dt\\
j_2 &= j_6 + j_7 + j_8 + j_9 &&= C_2 dv_2/dt\\
j_3 &= -j_6 - j_7 - j_8 - j_{10} &&= C_3 dv_3/dt\\
j_4 &= j_9 - j_{10} &&= v_4/R_4\\
j_5 &= j_9 - j_{10} &&= \int v_5 dt/L_5
\end{aligned}$$

$$\text{KVL}\quad\begin{aligned}
v_6 &= -v_2 + v_3 &&= R_6 j_6\\
v_7 &= -v_2 + v_3 &&= \int j_7 dt/C_7\\
v_8 &= -v_2 + v_3 &&= L_8 dj_8/dt\\
v_9 &= v_1 - v_2 - v_4 - v_5 &&= L_9 dj_9/dt\\
v_{10} &= -v_1 + v_3 + v_4 + v_5 &&= L_{10} dj_{10}/dt
\end{aligned}$$

$$\int v_5 dt/L_5 = j_9 - j_{10} \text{ and } \int j_7 dt/C_7 = -v_2 + v_3$$

are differentiated to produce

$$v_5/L_5 = v_9/L_9 - v_{10}/L_{10} \text{ and}$$

$$j_7/C_7 = -j_2/C_2 + j_3/C_3 \text{ respectively.}$$

v_5 and j_7 are then eliminated from the other equations which become

$$j_1 = -j_9 + j_{10}$$

$$j_2 = (2-C_2C_3/\Delta_7)j_6 + (C_2C_3/\Delta_7)j_8 + (1-C_3C_7/\Delta_7)j_9 -(C_2C_7/\Delta_7)j_{10}$$

$$j_3 = -(2-C_2C_3/\Delta_7)j_6 -(C_2C_3/\Delta_7)j_8 + (C_3C_7/\Delta_7)j_9 -(1-C_2C_7/\Delta_7)j_{10}$$

$$j_4 = j_9 - j_{10}$$

and

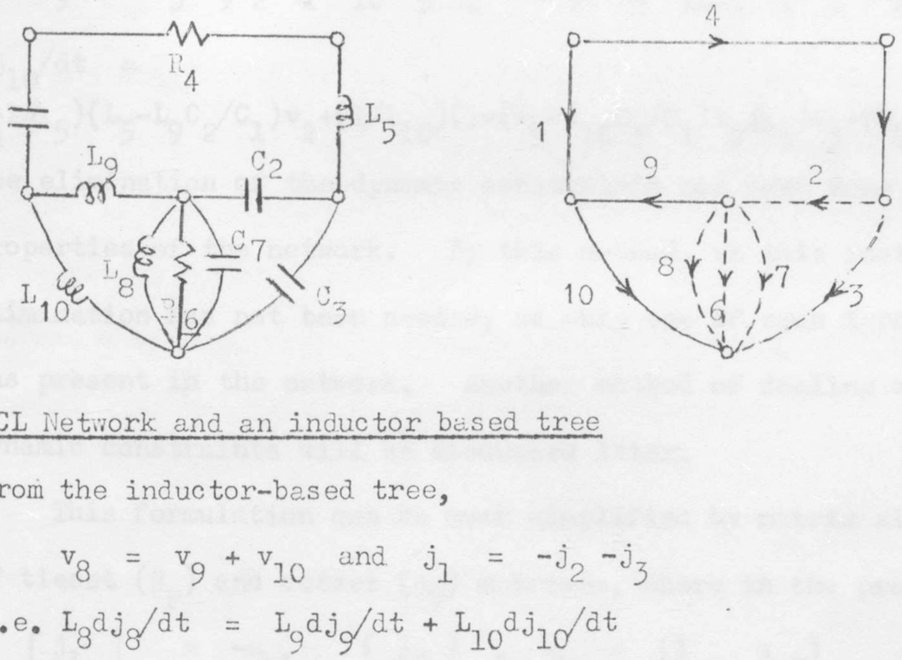
$$v_6 = -v_2 + v_3$$

$$v_8 = -v_2 + v_3$$

$$v_9 = (L_9L_{10}/\Delta_5)v_1 -(1-L_5L_{10}/\Delta_5)v_2 + (L_5L_9/\Delta_5)v_3 -(L_9L_{10}/\Delta_5)v_4$$

$$v_{10} = -(L_9L_{10}/\Delta_5)v_1 -(L_5L_{10}/\Delta_5)v_2 + (1-L_5L_9/\Delta_5)v_3 + (L_9L_{10}/\Delta_5)v_4$$

where $\Delta_7 = C_2C_3 + C_2C_7 + C_3C_7$ and $\Delta_5 = L_5L_9 + L_5L_{10} + L_9L_{10}$.



RCL Network and an inductor based tree

From the inductor-based tree,

$$v_8 = v_9 + v_{10} \quad \text{and} \quad j_1 = -j_2 - j_3$$

i.e. $L_8 dj_8/dt = L_9 dj_9/dt + L_{10} dj_{10}/dt$

and $C_1 dv_1/dt = -C_2 dv_2/dt - C_3 dv_3/dt$. *

Integrating and neglecting arbitrary constants

$$L_8 j_8 = L_9 j_9 + L_{10} j_{10} \quad \text{and} \quad C_1 v_1 = -C_2 v_2 - C_3 v_3$$

j_8 and v_1 are eliminated from the other equations which become

$$j_2 = (2 - C_2C_3/\Delta_7)j_6 + (1-(C_7 - C_2L_9/L_8)C_3/\Delta_7)j_9 + (C_2/\Delta_7)(-C_7 + C_3L_{10}/L_8)j_{10}$$

$$j_3 = -(2-C_2C_3/\Delta_7)j_6 + (C_3/\Delta_7)(C_7 - C_2L_9/L_8)j_9 -(1-(C_7-C_3L_{10}/L_8)C_2/\Delta_7)j_{10}$$

$$j_4 = j_9 - j_{10}$$

and

* which are to be added later as initial state variables.

$$v_6 = -v_2 + v_3$$

$$v_9 = -(1-(L_5-L_9C_2/C_1)L_{10}/\Delta_5)v_2 + (L_{10}/\Delta_5)(L_5-L_{10}C_3/C_1)v_3 - (L_9L_{10}/\Delta_5)v_4$$

$$v_{10} = -((L_5-L_9C_2/C_1)L_{10}/\Delta_5)v_2 + (1-(L_5-L_{10}C_3/C_1)L_9/\Delta_5)v_3 + (L_9L_{10}/\Delta_5)v_4$$

Finally the resistor equations

$$j_4 = v_4/R_4 = j_9 - j_{10} \text{ and } v_6 = R_6j_6 = -v_2 + v_3$$

are used to eliminate v_4 and j_6 from the other equations to produce

$$dv_2/dt =$$
$$(1/C_2R_6)(2-C_2C_3/\Delta_7)(-v_2+v_3)+(1/C_2)(1-(C_7-C_2L_9/L_8)C_3/\Delta_7)j_9+(1/\Delta_7)(-C_7+C_3L_{10}/L_8)j_{10}$$

$$dv_3/dt =$$
$$(1/C_3R_6)(2-C_2C_3/\Delta_7)(-v_2+v_3)+(1/\Delta_7)(C_7-C_2L_9/L_8)j_9-(1/C_3)(1-(C_7-C_3L_{10}/L_8)C_2/\Delta_7)j_{10}$$

$$dj_9/dt =$$
$$(-1/L_9)(1-(L_5-L_9C_2/C_1)L_{10}/\Delta_5)v_2+(1/\Delta_5)(L_5-L_{10}C_3/C_1)v_3-(R_4L_{10}/\Delta_5)(j_9-j_{10})$$

$$dj_{10}/dt =$$
$$(-1/\Delta_5)(L_5-L_9C_2/C_1)v_2+(1/L_{10})(1-(L_5-L_{10}C_3/C_1)L_9/\Delta_5)v_3+(R_4L_9/\Delta_5)(j_9-j_{10})$$

The elimination of the dynamic constraints has been done using topological properties of the network. By this method, in this instance, Gaussian Elimination has not been needed, as only one of each type of dynamic constraint was present in the network. Another method of dealing with the elimination of dynamic constraints will be discussed later.

This formulation can be much simplified by matrix algebra and the concepts of tieset (β_F) and cutset (α_F) matrices, where in the previous example

$$\begin{bmatrix} j_1 \\ j_2 \\ j_3 \\ j_4 \\ j_5 \end{bmatrix} = -\alpha_{nl} \begin{bmatrix} j_6 \\ j_7 \\ j_8 \\ j_9 \\ j_{10} \end{bmatrix}, \quad \alpha_F = (I_{nn} \quad \alpha_{nl})$$

n and l are the numbers of tree branches and links resp.

$n = l = 5$

I_{nn} is a unit submatrix of order n X n

α_{nl} is a submatrix of order n X l.

$$\begin{bmatrix} v_6 \\ v_7 \\ v_8 \\ v_9 \\ v_{10} \end{bmatrix} = -\beta_{ln} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{bmatrix}, \quad \beta_F = (\beta_{ln} \quad I_{ll})$$

I_{ll} is a unit submatrix of order l X l

β_{ln} is a submatrix of order l X n

$$\alpha_{nl} = \begin{bmatrix} 0 & 0 & 0 & 1 & -1 \\ -1 & -1 & -1 & -1 & 0 \\ 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix}$$

and

$$\beta_{ln} = \begin{bmatrix} 0 & 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ -1 & 1 & 0 & 1 & 1 \\ 1 & 0 & -1 & -1 & -1 \end{bmatrix}$$

Also $\alpha_{nl} + \beta_{ln}^T = \emptyset$ or $\alpha_{nl}^T + \beta_{ln} = \emptyset$, where α_{nl}^T and β_{ln}^T are the transposes of the matrices α_{nl} and β_{ln} respectively, so that only the tieset or cutset matrix needs to be determined. For further details see section 3.

The complexity of the equations pertaining to this simple RCL network suggests that for a really complex circuit, calculations by hand would be extremely tedious if not impossible. The computation is basically simple though lengthy, consequently a 'human computer' is likely to introduce errors through boredom, whereas a machine when programmed will work much faster and any errors made in general will be due to the programmer or the user.

2.2 Formulation of a Degenerate System of Equations[5]

For every network there exists many sets of independent variables in terms of which all other variables may be expressed. There are also sets of independent equations which completely describe the network in terms of the independent variables. A systematic method is required to identify these independent variables and formulate their corresponding Kirchhoff equations. Of the many sets of variables which can be chosen, there are three main types:-

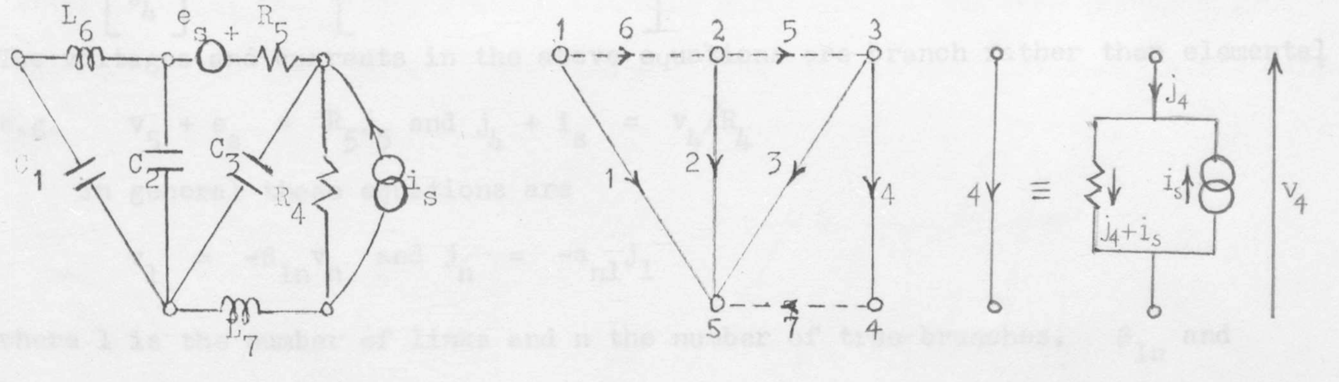
(i) node voltages,

(ii) mesh currents (or combinations of these two types called mixed methods),

(iii) capacitor voltages and inductor currents.

Topology provides us with a systematic method of choosing an independent set of variables and formulating the set of distinct equations which expresses the dependent variables in terms of the independent ones. The first step consists of drawing a set of branches connecting all nodes in a network but containing no loops. This set of branches is termed a tree.

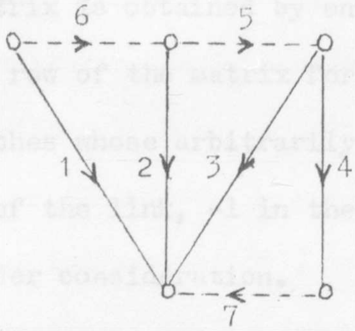
e.g.



Network with one particular tree

The branches forming the tree are called tree branches (designated by solid lines). The remaining branches are called links (designated by broken lines). There are of course many possible trees for a single network. The next step is to form sets of Kirchhoff Current and Kirchhoff Voltage Equations. The properties of the tree are utilised for this task. Each link completes a loop with a particular set of tree branches and a voltage equation can be written for each loop. These loops are termed tiesets. Also for each tree branch a current equation can be written relating it to some or all of the links but not to any of the other tree branches. The tree branch and its related links is termed a cutset.

e.g.



for the links we have

$$\begin{bmatrix} v_5 \\ v_6 \\ v_7 \end{bmatrix} = \begin{bmatrix} 0 & 1 & -1 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix}$$

and for the tree branches

$$\begin{bmatrix} j_1 \\ j_2 \\ j_3 \\ j_4 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} j_5 \\ j_6 \\ j_7 \end{bmatrix}$$

The voltages and currents in the above equations are branch rather than elemental

e.g. $v_5 + e_s = R_5 j_5$ and $j_4 + i_s = v_4/R_4$

In general these equations are

$$v_l = -\beta_{ln} v_n \quad \text{and} \quad j_n = -\alpha_{nl} j_l$$

where l is the number of links and n the number of tree branches. β_{ln} and α_{nl} are the subtieset and subcutset matrices of orders l x n and n x l respectively, where

$$\beta_F = (\beta_{ln} \quad I_{ll}) \text{ and } \alpha_F = (I_{nn} \quad \alpha_{nl}).$$

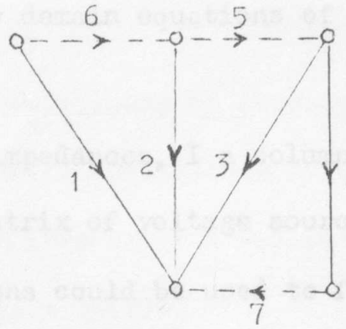
β_F and α_F are called the fundamental tieset and cutset matrices respectively. I_{ll} and I_{nn} are unit matrices of orders l x l and n x n resp.

Also β_{ln} and α_{nl} are related by the formula $\alpha_{nl} + \beta_{ln}^T = \emptyset$ where T means transpose.

The tieset matrix is obtained by entering the links one at a time in the tree and by forming one row of the matrix for each link by going around the tieset loop writing 1 for branches whose arbitrarily chosen current direction is in the same direction as that of the link, -1 in the opposite sense and Ø if not contained in the tieset loop under consideration.

e.g. tree branches

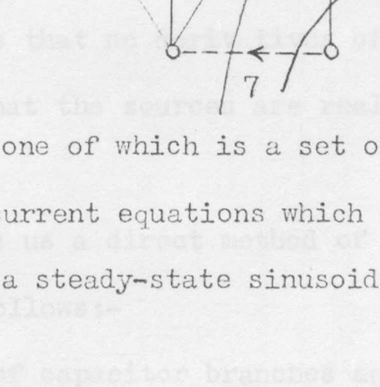
$$\text{links}\quad \begin{array}{c} \\ 5 \\ 6 \\ 7 \end{array}\begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \left[\begin{array}{cccc} 0 & -1 & 1 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 \end{array}\right] \end{array} = \beta_{ln}$$



The cutset matrix is usually obtained using the relationship $\alpha_{nl} + \beta_{ln}^{T} = 0$, but can be derived by drawing a line through the graph of the circuit so that it cuts one and only one tree branch and as many links as possible once and once only. Then a row of the cutset matrix is entered for each tree branch. For links entering the cutset line in the same direction as the tree branch a 1 is entered, -1 in the opposite sense and Ø if the link is not contained in the cutset.

e.g. links

$$\begin{array}{l} \\ \text{tree} \\ \text{branches} \\ \\ \end{array}\quad \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array}\begin{array}{c} \begin{array}{ccc} 5 & 6 & 7 \end{array} \\ \left[\begin{array}{ccc} 0 & 1 & 0 \\ 1 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 0 & -1 \end{array}\right] \end{array} = \alpha_{nl}$$



This results in two sets of equations, one of which is a set of tree branch voltage equations, the other a set of link current equations which could also be interpreted as mesh current equations. In a steady-state sinusoidal excitation Nodal Analysis equations of the form

$$Y * V = I_s$$

are sought, where Y is a matrix of complex admittances, V a column matrix of independent nodal voltages and I_s a column matrix of current sources. We could use the set of equations $\underline{v}_l = -\beta_{ln}\underline{v}_n$ along with $\underline{j}_l = (-\alpha_{nl})^{-1}\underline{j}_n$ in combination with $j_C = i\omega C v_C$, $j_L = (1/i\omega L)v_L$, $j_R = (1/R)v_R$ to solve for the dependent variables $(\underline{v}_l)$ in terms of the independent $(\underline{v}_n)$ to produce the required matrix system. Similarly for Mesh Analysis in the frequency demain equations of the form

$$Z * I = E_S$$

are sought, where Z is a matrix of complex impedances, I a column matrix of independent mesh currents and E_S a column matrix of voltage sources. Again the link current and tree branch voltage equations could be used to form the required matrix system, but in this case utilising the set of equations $\underline{j}_n = -\alpha_{nl}\underline{j}_l$ along with $\underline{v}_n = (-\beta_{ln})^{-1}\underline{v}_l$ in combination with $v_C = (1/i\omega C)j_C$, $v_R = Rj_R$, $v_L = i\omega L j_L$, to solve for the dependent variables $(\underline{j}_n)$ in terms of the independent $(\underline{j}_l)$.

For Nodal, Mesh and Mixed Analyses in the time domain using the relations $j_C(t) = C dv_C(t)/dt$, $v_R(t) = Rj_R(t)$, $v_L(t) = L dj_L(t)/dt$ to express the dependent variables in terms of the independent variables leads in general to mixed systems of algebraic, integral and differential equations. State Variable Methods strive to produce matrix systems of independent first order differential equations of the form

$$d\underline{x}(t)/dt = A\underline{x}(t) + B\underline{u}(t)$$

where $\underline{x}(t)$ is a vector of state variables and $\underline{u}(t)$ a vector of excitations to the circuit represented by these equations. Choosing capacitor voltages and inductor currents as state variables ensures that no derivatives of the excitations $\underline{u}(t)$ can appear.[1] (Provided that the sources are real i.e. they have an internal resistance.)

Bashkow's Topological Formulation gives us a direct method of producing these state equations. The method is as follows:-

(1) Choose a tree with the maximum number of capacitor branches and the minimum number of inductor branches.

(2) Form the appropriate tieset and cutset matrices.

(3) Form the degenerate system of equations

$$\begin{bmatrix} \underline{j}_n \\ \underline{v}_l \end{bmatrix} = \begin{bmatrix} 0_{nn} & -\alpha_{nl} \\ -\beta_{ln} & 0_{ll} \end{bmatrix} \begin{bmatrix} \underline{v}_n \\ \underline{j}_l \end{bmatrix} = \begin{bmatrix} 0_{nn} & \beta_{ln}^T \\ -\beta_{ln} & 0_{ll} \end{bmatrix} \begin{bmatrix} \underline{v}_n \\ \underline{j}_l \end{bmatrix}$$

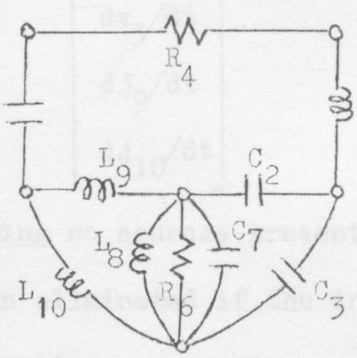
(4) Eliminate any capacitor variables occurring in the links and any inductor variables occurring in the tree branches.

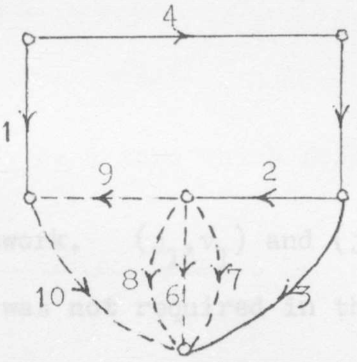
(5) Eliminate any excess inductor variables introduced by loops of inductors and any excess capacitor variables due to dynamic constraints between capacitor currents. This part could be omitted if the inverse of A was not required in the solution of the equations $d\underline{x}(t)/dt = A\underline{x}(t) + B\underline{u}(t)$.

(6) Eliminate any resistor variables and bring the remaining equations into the form

$$d\underline{x}(t)/dt = A\underline{x}(t) + B\underline{u}(t).$$

e.g.





$$\beta_{ln} = \begin{bmatrix} 0 & 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ -1 & 1 & 0 & 1 & 1 \\ 1 & 0 & -1 & -1 & -1 \end{bmatrix}$$

$$\begin{bmatrix} j_1 \\ j_2 \\ j_3 \\ j_4 \\ j_5 \\ v_6 \\ v_7 \\ v_8 \\ v_9 \\ v_{10} \end{bmatrix} = \begin{bmatrix} & & 0_{nn} & & & 0 & 0 & 0 & -1 & 1 \\ & & & & & 1 & 1 & 1 & 1 & 0 \\ & & & & & -1 & -1 & -1 & 0 & -1 \\ & & & & & 0 & 0 & 0 & 1 & -1 \\ & & & & & 0 & 0 & 0 & 1 & -1 \\ 0 & -1 & 1 & 0 & 0 & & & & & \\ 0 & -1 & 1 & 0 & 0 & & & & & \\ 0 & -1 & 1 & 0 & 0 & & & 0_{ll} & & \\ 1 & -1 & 0 & -1 & -1 & & & & & \\ -1 & 0 & 1 & 1 & 1 & & & & & \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ j_6 \\ j_7 \\ j_8 \\ j_9 \\ j_{10} \end{bmatrix}$$

and
$$\begin{bmatrix} j_1 \\ j_2 \\ j_3 \\ j_4 \\ j_5 \\ v_6 \\ v_7 \\ v_8 \\ v_9 \\ v_{10} \end{bmatrix} = \begin{bmatrix} C_1 dv_1/dt \\ C_2 dv_2/dt \\ C_3 dv_3/dt \\ v_4/R_4 \\ \int v_5 dt/L_5 \\ j_6/R_6 \\ \int j_7 dt/C_7 \\ L_8 dj_8/dt \\ L_9 dj_9/dt \\ L_{10} dj_{10}/dt \end{bmatrix}$$

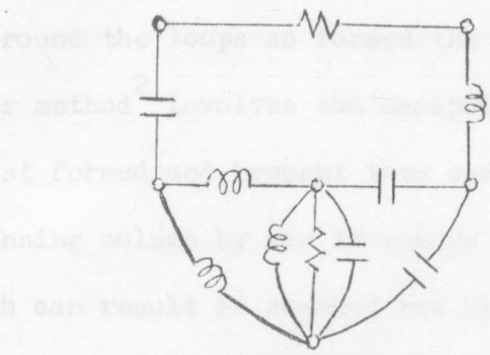
$(j_5,v_5),(j_7,v_7),(j_1,v_1),(j_8,v_8),(j_4,v_4),(j_6,v_6)$ are eliminated leaving behind a system of the form

$$\begin{bmatrix} dv_2/dt \\ dv_3/dt \\ dj_9/dt \\ dj_{10}/dt \end{bmatrix} = A \begin{bmatrix} v_2 \\ v_3 \\ j_9 \\ j_{10} \end{bmatrix}$$

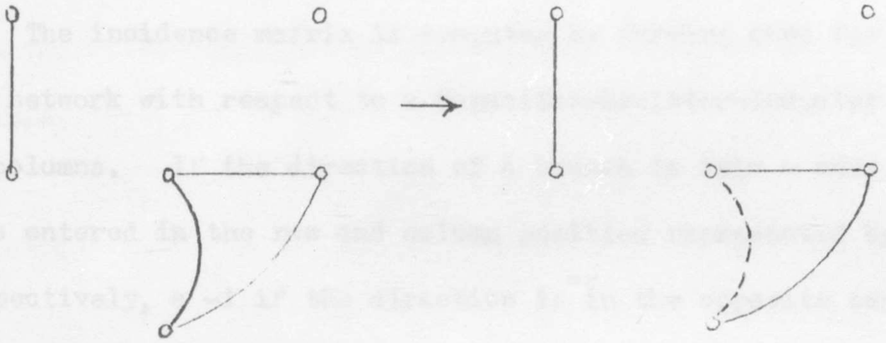
there being no sources present in the network. (j_1,v_1) and (j_8,v_8) need not have been eliminated if the inverse of A was not required in the solution of these equations.

Bashkow's Topological Formulation lends itself to computer implementation. The stages in formulation remain much the same but the computation of an optimum tree along with its associated tieset and cutset matrices are slightly different from that done by hand. There are two main methods of computing a tree. One is a Search Method which involves the scanning of branches (entered one type at a time) for loops and taking appropriate action when the criteria for identifying a tree have been fulfilled.

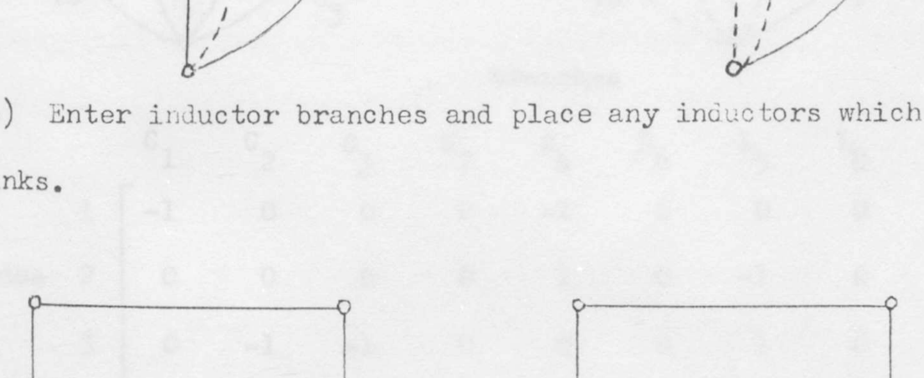
e.g.



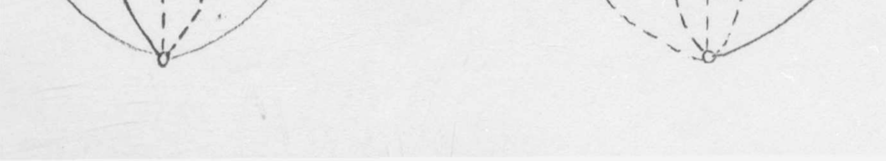
(1) Enter capacitor branches and place any excess capacitors in the links.



(2) Enter resistor branches and place any resistors which form loops into the links.



(3) Enter inductor branches and place any inductors which form loops into the links.

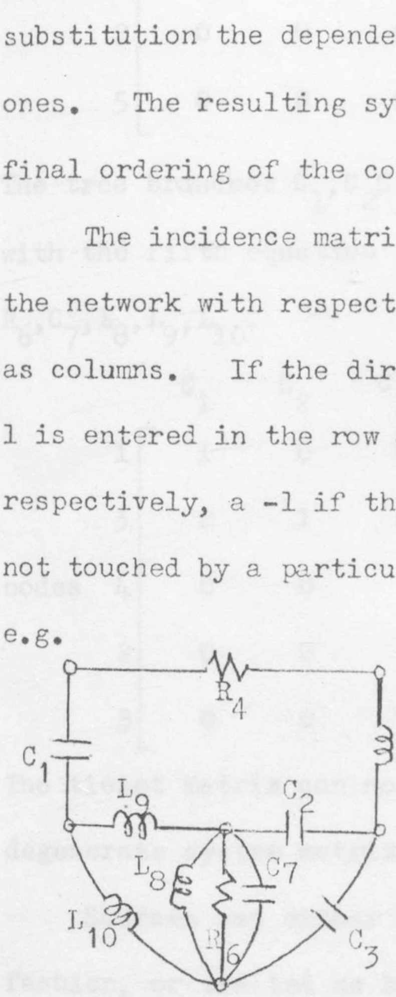


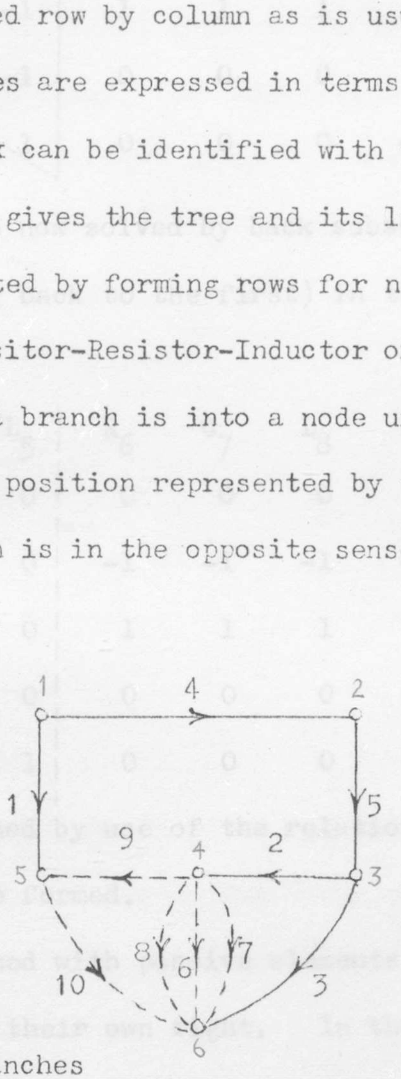
The tree is now complete as all nodes are connected and no loops are formed by tree branches. Now placing the links into the tree one and only one at a time and scanning round the loops so formed the tieset matrix can be computed.

The other method^2 involves the manipulation of matrices. The Incidence Matrix is first formed and brought into row-echelon form by a Gaussian Full Pivot technique scanning column by row to ensure that there are no unnecessary column changes, which can result if scanned row by column as is usually done. By back substitution the dependent variables are expressed in terms of the independent ones. The resulting system matrix can be identified with the cutset matrix. The final ordering of the columns also gives the tree and its links.

The incidence matrix is computed by forming rows for n of the n + 1 nodes of the network with respect to a Capacitor-Resistor-Inductor ordering of the branches as columns. If the direction of a branch is into a node under consideration a 1 is entered in the row and column position represented by the node and branch respectively, a -1 if the direction is in the opposite sense and Ø if the node is not touched by a particular branch.

e.g.





branches

$$\begin{array}{cc|cccccccccc|}
 & & C_1 & C_2 & C_3 & C_7 & R_4 & R_6 & L_5 & L_8 & L_9 & L_{10} \\
 & 1 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\
\text{nodes} & 2 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 \\
 & 3 & 0 & -1 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
 & 4 & 0 & 1 & 0 & -1 & 0 & -1 & 0 & -1 & -1 & 0 \\
 & 5 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1
\end{array}$$

The incidence matrix is now brought into row-echelon form (also known as upper triangular form).

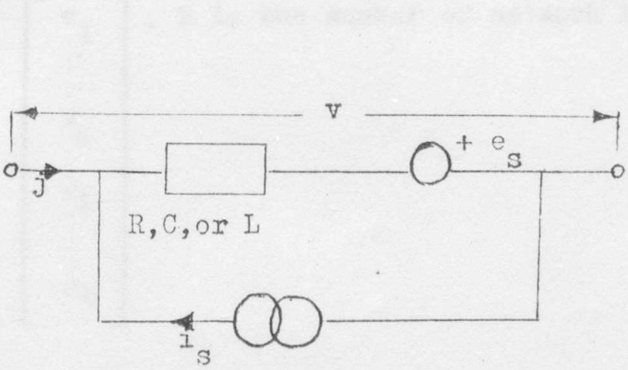
$$\begin{array}{cc} & \longleftarrow \text{tree branches} \longrightarrow \longleftarrow \text{links} \longrightarrow \\ & \begin{array}{cccccccccc} C_1 & C_2 & C_3 & R_4 & L_5 & R_6 & C_7 & L_8 & L_9 & L_{10} \end{array} \\ \text{nodes} \begin{array}{c} 1 \\ 3 \\ 4 \\ 2 \\ 5 \end{array} & \left[\begin{array}{cccccccccc} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 1 \end{array}\right] \end{array}$$

The tree branches $C_1, C_2 C_3, R_4, L_5$ are now solved by back substitution (starting with the fifth equation and working back to the first) in terms of the links $R_6, C_7, L_8, L_9, L_{10}$.

$$\begin{array}{cc} & \begin{array}{ccccc|ccccc} C_1 & C_2 & C_3 & R_4 & L_5 & R_6 & C_7 & L_8 & L_9 & L_{10} \end{array} \\ \text{nodes} \begin{array}{c} 1 \\ 3 \\ 4 \\ 2 \\ 5 \end{array} & \left[\begin{array}{ccccc|ccccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 1 & 0 & 0 & 0 & -1 & -1 & -1 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 1 \end{array}\right] \end{array} = (I_{nn} \quad \alpha_{nl})$$

The tieset matrix can now be obtained by use of the relationship $\beta_{ln} = -\alpha_{nl}^T$. The degenerate system matrix can now be formed.

Sources can either be associated with passive elements in some consistent fashion, or treated as branches in their own right. In the former case a compound branch is formed



$v + e_s = R(j + i_s)$, $v + e_s = Ld(j + i_s)/dt$, or $j + i_s = Cd(v + e_s)/dt$.

It can be seen that this method of source representation in general introduces first derivatives of the sources which is something we wish to avoid. The problem can be avoided by placing a small resistance in series with an ideal voltage source, a large resistance in parallel with an ideal current sourse and associating the sources with these resistances. In the latter case first derivatives of the sources can appear if the sources are ideal, though this would occur later in the formulation during the elimination of any dynamic constraints. This will be discussed in more detail in the next section. The initial order of the branches in the independent method needs some modification. The initial order was - capacitors followed by resistors followed by inductors. Now the order of preference must be as follows:-

Dependent Voltage Sources

Independent Voltage Sources

Capacitors

Resistors

Inductors

Independent Current Sources

Dependent Current Sources.

After the tree has been chosen some of the branches may not be in this order which signifies the presence of constraints. During the formulation using either method of source representation, the dependent sources can be treated (for the most part) as if they were independent.

If output variables y are required, they can be entered initially in the form

$$\underline{y} = c \begin{bmatrix} v_1 \\ \vdots \\ v_b \\ j_1 \\ \vdots \\ j_b \end{bmatrix}$$

, b is the number of network branches.

When a tree has been chosen, the column matrix containing the branch voltages and currents could be arranged so that

$$\underline{y} = \begin{bmatrix} C_1 & C_2 & C_3 & C_4 \end{bmatrix} \begin{bmatrix} \underline{v}_n \\ \underline{v}_l \\ \underline{j}_n \\ \underline{j}_l \end{bmatrix}$$

where $\underline{v}_n$ and $\underline{j}_n$ are the tree branch voltages and currents, $\underline{v}_l$ and $\underline{j}_l$ are the link voltages and currents.

Using the topological relations

$$\underline{j}_n = \beta_{ln}^T \underline{j}_l \quad \text{and} \quad \underline{v}_l = -\beta_{ln} \underline{v}_n$$

the output equations become

$$\underline{y} = \begin{bmatrix} C_1 - C_2 \beta_{ln} & \vdots & C_4 + C_3 \beta_{ln}^T \end{bmatrix} \begin{bmatrix} \underline{v}_n \\ \underline{j}_l \end{bmatrix}$$

and can now be updated simultaneously with

$$\begin{bmatrix} \underline{j}_n \\ \underline{v}_l \end{bmatrix} = A \begin{bmatrix} \underline{v}_n \\ \underline{j}_l \end{bmatrix}$$

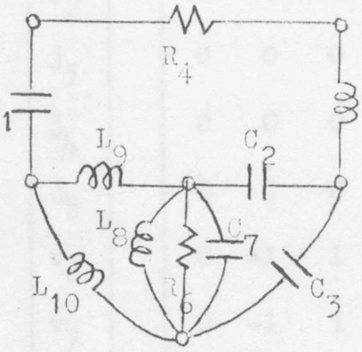
during the formulation process.

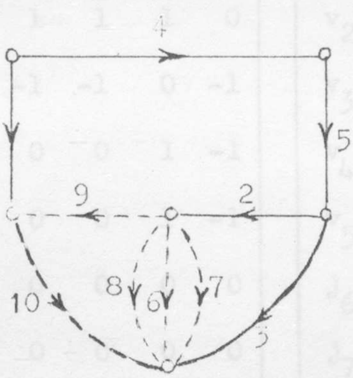
2.3 Constraints[5]

There are two types of constraint, dynamic and non-dynamic. The former affect the order (number of independent energy storage devices) of the network, the latter do not. In an RCL circuit with real sources, there are four possible types of dynamic constraints: tiesets of capacitors, cutsets of inductors, loops of inductors, and capacitor current constraints. The possible non-dynamic constraints are loops of resistors, and resistor current constraints. If the sources are not real, several other types of dynamic constraints are possible. Any tiesets containing capacitors, independent voltage sources, or capacitor voltage controlled voltage sources reduce the order of the system, as do cutsets with inductors in combination with independent current sources, or inductor current controlled current sources; these two types of constraints can introduce first derivatives of the sources into the formulation. Integrals of the sources can also be introduced by constraints formed from loops containing inductors, independent voltage sources, and inductor voltage controlled voltage sources, and by capacitor current constraints with independent current sources or capacitor current controlled current sources. Inconsistencies can also appear. Loops of independent voltage sources are inconsistent as they cannot be independent. Some loops of dependent voltage sources are consistent in the sense that they do not produce contradictory relations between variables. Similarly, cutsets of independent current sources are inconsistent but cutsets of dependent current sources are not necessarily inconsistent. Any derivatives of the sources may be eliminated from the state equations by a suitable transformation of the state vector, but not necessarily from the output-state equations, so that some numerical differentiation may be required during the computation of prescribed output variables. Therefore keeping the sources real greatly simplifies the classification of the constraints and their elimination, though it may introduce computational errors, and could slow down the solution process.

Associating the sources with a resistor, capacitor tiesets and inductor cutsets can be enumerated and their equations of constraint formulated simultaneously by drawing a capacitor based tree for the network i.e. one with as many capacitor branches in the tree and inductor branches in the links as possible. Any inductor cutsets will appear in the tree branch equations and any capacitor tiesets in the link equations. Similarly trees can be drawn to enumerate and formulate the other equations of constraint. For instance, inductor loop and capacitor current constraints are dealt with by drawing a tree with as many inductors in the tree and capacitors in the links as possible. The capacitor cutsets appear in the tree branch equations and the inductor tiesets in the link equations. Using the same routines the resistor tiesets are found using a resistor based tree; the resistor cutsets using a non-resistor based tree.

EXAMPLE

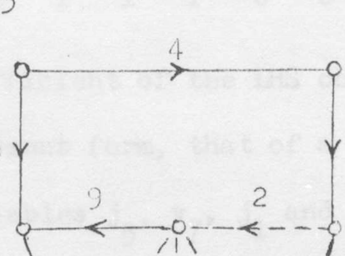




capacitor based tree

$j_5 \;=\; j_9 - j_{10}$ inductor cutset

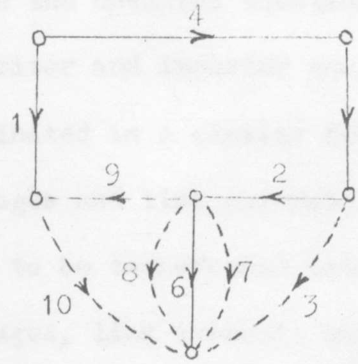
$v_7 \;=\; -v_2 + v_3$ capacitor tieset



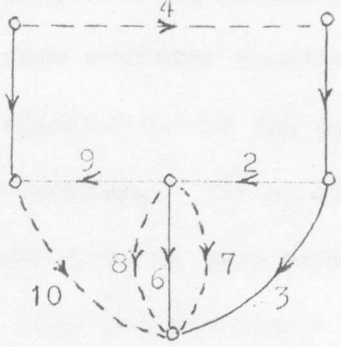
inductor based tree

$j_1 = -j_2 - j_3$ capacitor cutset

$v_8 = v_9 + v_{10}$ inductor tieset



resistor based tree



non-resistor based tree

There are no resistor constraint equations. These constraint equations are now used to eliminate the pairs of variables (j_5, v_5), (j_7, v_7), (j_1, v_1) and (j_8, v_8) from the degenerate system of equations.

$$I \begin{bmatrix} j_1 \\ j_2 \\ j_3 \\ j_4 \\ j_5 \\ v_6 \\ v_7 \\ v_8 \\ v_9 \\ v_{10} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & -1 & -1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ j_6 \\ j_7 \\ j_8 \\ j_9 \\ j_{10} \end{bmatrix}$$

If the matrix coefficient of the LHS column matrix of this system of equations is kept in its present form, that of a unit matrix, this will effectively eliminate the variables j_5, v_7, j_1 and v_8. The procedure is to first dispose of the variables v_5, j_7, v_1 and j_8, keeping the matrix coefficient of the LHS column matrix as unity. The equations which correspond to j_5, v_7, j_1 and v_8 in the LHS column matrix are then discarded. In general four subsystems of equations of dynamic constraints would have to be dealt with. These would be solved by Gaussian Elimination and the dependent tree branch voltages and link currents substituted back in the main system of degenerate equations, after which the unwanted equations are discarded, leaving behind a set of independent capacitor and inductor equations, plus some resistor equations which can be eliminated in a similar fashion. The elimination of the dependent tree branch voltages and link currents can be quite complex. The equations of constraint have to be transformed using the relations between tree branch currents and voltages, link currents and voltages. This unfortunately introduces more tree branch currents and link voltages into the capacitor tieset and inductor cutset constraint equations. At this stage any tree branch currents and link voltages are eliminated by referring back to the main system of degenerate equations. The constraint subsystems now involve only tree branch voltages and link currents, but are interactive with each other and with the main system of equations. This complicates even more the elimination of any unwanted variables. The following procedure has been adopted:

(1) The relevant trees and tieset matrices are computed.

(2) The equations of constraint are picked out and transformed so that they contain only tree branch voltages and link currents.

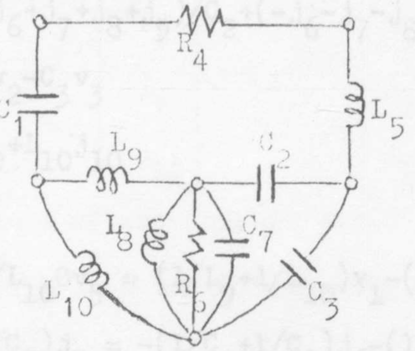
(3) The capacitor equations of constraint are solved by Gaussian Elimination, and the dependent variables substituted back into the main system, the inductor constraint subsystems and the resistor constraint subsystems.

(4) The inductor equations of constraint are solved in the same way, and the dependent variables substituted back into the main system and the resistor constraint subsystems.

(5) The resistor equations of constraint are solved and the dependent variables substituted back into the main system.

(6) Any remaining resistor equations in the main system are solved and substituted back into the remaining capacitor and inductor equations which are then brought into State Variable form.

EXAMPLE



The equations of constraint are as before

$j_5 = j_9 - j_{10}$ inductor cutset

$v_7 = -v_2 + v_3$ capacitor tieset

$j_1 = -j_2 - j_3$ capacitor cutset

$v_8 = v_9 + v_{10}$ inductor tieset

which become

$$\left.\begin{aligned} v_5/L_5 &= v_9/L_9 - v_{10}/L_{10} \\ j_7/C_7 &= -j_2/C_2 + j_3/C_3 \\ C_1 v_1 &= -C_2 v_2 - C_3 v_3 \\ L_8 j_8 &= L_9 j_9 + L_{10} j_{10} \end{aligned}\right\} \text{neglecting arbitrary constants of integration.}^{*}$$

Referring back to the main system

$$\begin{bmatrix} j_1 \\ j_2 \\ j_3 \\ j_4 \\ j_5 \\ v_6 \\ v_7 \\ v_8 \\ v_9 \\ v_{10} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & -1 & -1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ j_6 \\ j_7 \\ j_8 \\ j_9 \\ j_{10} \end{bmatrix}$$

the equations of constraint become

* which are to be added later as initial state variables.

$$v_5/L_5 = (v_1-v_2-v_4-v_5)/L_9-(-v_1+v_3+v_4+v_5)/L_{10}$$

$$j_7/C_7 = -(j_6+j_7+j_8+j_9)/C_2+(-j_6-j_7-j_8-j_{10})/C_3$$

$$C_1v_1 = -C_2v_2-C_3v_3$$

$$L_8j_8 = L_9j_9+L_{10}j_{10}$$

i.e.

$$(1/L_5+1/L_9+1/L_{10}\text{O}v_5 = (1/L_9+1/L_{10})v_1-(1/L_9)v_2 \quad -(1/L_{10})v_3-(1/L_9+1/L_{10})v_4$$

$$(1/C_2+1/C_3+1/C_7)j_7 = -(1/C_2+1/C_3)j_6-(1/C_2+1/C_3)j_8-(1/C_2)j_9-(1/C_3)j_{10}$$

$$C_1v_1 = -C_2v_2-C_3v_3$$

$$L_8j_8 = L_9j_9+L_{10}j_{10}$$

Eliminating j_7 and v_1 from the main system and from the other equations of constraint we are left with the constraint equations

$$(1/L_5+1/L_9+1/L_{10})v_5 = (1/L_9+1/L_{10})((-C_2v_2-C_3v_3)/C_1)-(1/L_9)v_2-(1/L_9+1/L_{10})v_4$$
$$-(1/L_{10})v_3$$

$$L_8j_8 = L_9j_9+L_{10}j_{10}$$

and after discarding the v_7 and j_1 equations a main system of the form

$$\begin{bmatrix} j_2 \\ j_3 \\ j_4 \\ j_5 \\ v_6 \\ v_8 \\ v_9 \\ v_{10} \end{bmatrix} = A \begin{bmatrix} v_2 \\ v_3 \\ v_4 \\ v_5 \\ j_6 \\ j_8 \\ j_9 \\ j_{10} \end{bmatrix}$$

We now substitute for v_5 and j_8 in the main system and discard the j_5 and v_8 equations. Finally the resistor variables v_4 and j_6 are determined and substituted back into the remaining equations which are brought into state variable form.

$$d/dt \begin{bmatrix} v_2 \\ v_3 \\ j_9 \\ j_{10} \end{bmatrix} = A \begin{bmatrix} v_2 \\ v_3 \\ j_9 \\ j_{10} \end{bmatrix}$$

The presence of any sources would mean the progressive updating of two matrices A and B such that eventually we arrive at equations of the form

$$d\underline{x}/dt = A\underline{x} + B\underline{u}$$

where $\underline{x}$ and $\underline{u}$ are the vectors of state variables and excitations respectively. If output variables $\underline{y}$ are prescribed two more matrices C and D require updating along with A and B so that output-state equations

$$\underline{y} = C\underline{x} + D\underline{u}$$

are produced.

The inductor tieset and capacitor cutset constraints need not be eliminated if the inverse of A is not required during the solution of the state equations. However if there are a lot of these constraints present, we would be solving a much larger system of equations than is really necessary which could considerably slow down the solution process.

Another method^2 of identifying and eliminating dynamic constraints is to use the tieset or cutset matrices to set up initial equations of the form

$$Sd/dt \begin{bmatrix} \underline{V}_c \\ \underline{v}_c \\ \underline{i}_L \\ \underline{I}_L \end{bmatrix} = A \begin{bmatrix} \underline{V}_c \\ \underline{v}_c \\ \underline{i}_L \\ \underline{I}_L \end{bmatrix} + B\underline{u}$$

where $\underline{i}_L$ are the excess inductor currents due to the presence of cutsets of inductors and sources; $\underline{v}_c$ are the excess capacitor voltages due to the presence of tiesets of capacitors and sources. The augmented matrix (S A B) is reduced to row-echelon form. If S has at least one row of zeros, but the corresponding row of A has at least one non-zero element, then a constraint exists between the state variables in this row, at least one of which can be eliminated together with its first derivative. After substituting for the dependent variables in terms of the remaining independent ones, the process is repeated until S becomes a diagonal matrix. If (S A) has a zero row, but the corresponding row of B has at least one non-zero element, then a constraint exists between the sources. If $\underline{u}$ contains only independent sources, then the network is inconsistent and has no solution. A constraint between controlled sources may be consistent. If (S A B) has a zero row this denotes the presence of a completely superfluous state equation. When S becomes a diagonal matrix, the equations could then be solved by some method which does not require an inverse of A which in general would not exist due to the possible presence of cutsets of capacitors and sources or tiesets of inductors and sources. These latter sets of constraints could be removed by reducing the matrix (A S B) to row-echelon form. This would reveal any constraints between first derivatives of state variables. After elimination of any unwanted variables, S would have to be converted to a unit matrix and the equations would be in their final form. Operating on (S A B) and eliminating state variables and their first derivatives introduces in general derivatives of the sources. Similarly operating on (A S B) and eliminating state variables and their first derivatives introduces in general integrals of the sources. These derivatives and integrals of sources should not occur if the sources are real.

2.4 Conclusions

Methods based on Bashkow's Topological Formulation provide a direct way of obtaining a set of independent first order differential equations, called the state equations, which together with all inputs completely describe a network.

There are two main methods of formulating the state equations. In one the tieset or cutset matrices are used directly to form a set of equations

$$\begin{bmatrix} j_1 \\ j_n \\ v_{n+1} \\ v_b \end{bmatrix} = \begin{bmatrix} 0_{nn} & \beta_{ln}^{T} \\ -\beta_{ln} & 0_{ll} \end{bmatrix} \begin{bmatrix} v_1 \\ v_n \\ j_{n+1} \\ j_b \end{bmatrix}$$

In this method, any capacitor tiesets or inductor cutsets would appear as integral equations after substituting for tree branch currents and link voltages. Any tree branch inductor voltages and currents or link capacitor voltages and currents are eliminated. Any resistor variables are also eliminated leaving behind a system of first order differential equations which are independent if the network represented by the equations contains no tiesets of inductors and sources or cutsets of capacitors and sources. The presence of these latter constraints could slow down the solution process considerably if the equations are solved without their removal. The equations of constraint can be formulated by drawing auxiliary trees, computing their associated tieset or cutset matrices, and selecting the appropriate rows. If these constraint equations are put in terms of tree branch voltages and link currents only, the effective elimination of the first derivatives of the selected dependent variables can be achieved by discarding the appropriate equations from the main system after the necessary substitutions have been made.

In the other method of formulating the state equations, the tieset or cutset matrices are first computed and then used to produce equations of the form

$$S\,d/dt\begin{bmatrix}\underline{V}_C\\ \underline{v}_C\\ \underline{i}_L\\ \underline{I}_L\end{bmatrix} = A\begin{bmatrix}\underline{V}_C\\ \underline{v}_C\\ \underline{i}_L\\ \underline{I}_L\end{bmatrix} + B\underline{u}$$

Where $\underline{v}_C$ and $\underline{i}_L$ are the excess capacitor voltages and inductor currents due to the presence of tiesets of capacitors and sources, and cutsets of inductors and sources respectively in the network being represented. These constraints are dealt with by repeated Gaussian Elimination on the augmented matrix (S A B) until S becomes a diagonal matrix. Any constraints pertaining to tiesets of inductors and sources or cutsets of capacitors and sources could be removed similarly by operating on (A S B) until A becomes diagonal, after which (S A B) would have to be operated on until S became a unit matrix. The latter method is in some respects more elegant in form than the former, but suffers from repeated Gaussian Elimination on what could be in practice a very large matrix and consequently the round-off error build up would have to be carefully monitored. It also needs in practice an independent method of finding the numbers of the different kinds of constraints.

If the sources are not real, derivatives of the sources could appear even when the state variables are capacitor voltages and inductor currents due to the presence of cutsets of inductors and sources or tieset of capacitors and sources; if constraints due to tiesets of inductors and sources or cutsets of capacitors and sources are to be removed this could introduce integrals of sources. Sources may be represented as branches in their own right or associated with a passive element. Derivatives or integrals of sources can be transformed out of the state equations by a suitable change of variable, but not in general from the output-state equations. The input vector $\underline{u}$ will in general contain independent and non-linear controlled sources.

3.0 Solution of the State Equations[2]

3.1 Introduction

For an nth order system with m real excitations, and capacitor voltages and inductor currents taken as state variables, the state equations take the form

$$d\underline{x}(t)/dt = A\underline{x}(t) + B\underline{u}(t) \qquad (3.1.1)$$

where $d\underline{x}/dt$, A, $\underline{x}$, B, $\underline{u}$ are matrices of orders $n \times 1$, $n \times n$, $n \times 1$, $n \times m$ and $m \times 1$ respectively. In general there will also be a set of k output-state equations

$$\underline{y}(t) = C\underline{x} + D\underline{u} \qquad (3.1.2)$$

where $\underline{y}$, C, D are matrices of orders $k \times 1$, $k \times n$, and $k \times m$. The value of k and the matrices C and D are prescribed by the user, though C and D will have been modified within the program. If capacitor voltages and inductor currents are not used as the state variables $\underline{x}$, derivatives of the inputs $\underline{u}$ may appear even though the sources are real.

(3.1.1) may be tackled by a predictor corrector method such as Adams-Bashforth-Moulton or a Runge-Kutta method such as England's method.[4] Alternatively the state equations can be expressed in the form

$$\underline{x}(t) = e^{At}\,\underline{x}(\emptyset) + \int_{\emptyset}^{t} e^{A(t-t')}\,B\underline{x}(t')dt \qquad (3.1.3.)$$

where $\underline{x}(\emptyset)$ is the initial state vector. The problem is virtually reduced to dealing with e^{At} and $\int e^{At}dt$, where

$$e^{At} = I + At + \cdots\cdots + A^p t^p/p! +$$

and I is an $n \times n$ unit matrix. If the input vector $\underline{u}$ can be regarded as constant over some interval $ph \leqslant t \leqslant (p+1)h$, (p positive integer) then (3.1.3) can be expressed

$$\underline{x}((p+1)h) = e^{Ah}\,\underline{x}(ph) + (e^{Ah} - I)A^{-1}B\underline{u}(ph)$$

$$= \phi\underline{x}(ph) + \psi\underline{u}(ph) \qquad (3.1.4)$$

This is sometimes known as Certaines' Method. For some problems ϕ and ψ can be evaluated once and for all at the beginning of the computation.

3.2 Certaine's Method

The linear difference equations (3.1.4) may be solved stage by stage using the previous state vector $\underline{x}(ph)$ to produce $\underline{x}((p+1)h)$. ϕ and ψ are re-evaluated as necessary according to some error tolerance criteria. A step size varying routine can also be incorporated which would re-evaluate ϕ and ψ whenever the step size is changed. The output-state vector would also be computed at each stage and output or stored for future use.

ϕ and ψ are known as the transition matrices and are calculated by truncating their series expansions after some convergence criteria have been satisfied. The transition matrices may be evaluated almost simultaneously by the relations

$$\phi = \sum_{m=0}^{\infty} A^m h^m / m! \quad \text{and} \quad \psi = h \left[\sum_{m=0}^{\infty} A^m h^m / (m+1)! \right] B.$$

As the matrix series for ϕ is built up, the (i, j) element ϕ_{ij} is incremented by $\delta\phi_{ij}$ to become $\phi_{ij} + \delta\phi_{ij}$. Eventually, all $|\delta\phi_{ij}/\phi_{ij}| < \varepsilon$ (some prescribed small positive number), and both infinite series for ϕ and ψ are truncated, as the latter converges more rapidly than the former. Unfortunately, if the number of terms in the infinite series for ϕ is more than 7 or 8 the method becomes numerically unstable. Therefore a step size h has to be chosen such that

$$\phi = \sum_{m=0}^{\infty} A^m h^m / m!$$

converges for m less than 8. This restriction has been found in practice to cause the solution process to almost grind to a halt, especially with networks which have slowly damped oscillatory responses.

The assumption of constant inputs $\underline{u}$ over some interval $pH \leqslant t < (p+1)h$ holds better than would be expected for inputs with piecewise-linear functions of time, but breaks down with non-linear inputs. The method also fails when two or more State variables have widely separated time constants, due to large Lipschitz constants being introduced which severely slow down the rate of convergence to a solution.

Other difference formulae can be produced to deal with the problems of sources not being constant over some small time interval and non-linear inputs, but the stability of the method is not improved and the appearance of widely separated time constants still wreaks havoc.

To deal with this latter problem known as stiffness we have to use more powerful methods of numerical analysis. In fact, no Runge-Kutta method or Predictor Corrector method can deal in general with this problem of stiffness and we have to turn to a special class of methods due to Gear, Krogh and others.

3.3 Gear's Method[4]

This method incorporates into a sophisticated program Adams-Bashforth-Moulton methods of orders one to seven. The program automatically changes the order of the method, as well as the step size, the strategy being to minimize the computational effort required to keep local errors within a certain tolerance. The program offers an option to the user of replacing these Adams-Bashforth-Moulton methods by special predictor-corrector methods appropriate to the solution of a stiff system of differential equations. Krogh's method is similar but has Adams-Bashforth-Moulton methods of orders one to thirteen available to the user.

The Adams-Bashforth pth order predictor equation for the differential equation $dx/dt = f(x,t)$ is

$$x_{n,(\emptyset)} = x_{n-1} + h \sum_{i=1}^{p} \beta_i d(x_{n-i})/dt \qquad (3.3.1)$$

where $x_i = x(ih)$, $dx_i/dt = f(x_i, ih)$, h is the step size. The sets of constants β_i for each method of order p can be found in almost any standard numerical analysis textbook. $x_{n,(\emptyset)}$ is the value predicted by (3.3.1) to be used as a first approximation of the Adams-Moulton corrector formula of order p

$$x_{n,(m+1)} = x_{n-1} + \beta^*_{\emptyset} hf(x_{n,(m)}, t_n) + h \sum_{i=1}^{p-1} \beta^*_i d(x_{n-i})/dt \qquad (3.3.2)$$

If the corrected value $x_{n,(m+1)}$ fails to converge in three iterations, the step size is reduced and a new predictor value computed from (3.3.1). The set of β^*_i for each method of order p can be found similarly to the β_i.[8] For stiff equations, the predictor and corrector equations are given by

$$x_{n,(\emptyset)} = \sum_{i=1}^{p} \alpha_i x_{n-i} + \eta_1 hd(x_{n-1})/dt \qquad (3.3.3)$$

and

$$x_{n,(m+1)} = \sum_{i=1}^{p} \alpha^*_i x_{n-i} + \eta^*_{\emptyset} hf(x_{n,(m)}, t_n) \qquad (3.3.4)$$

where the coefficients α_i, α^*_i, $\eta^*_{\emptyset}$, η_1 are given in [7].

The above equations can be expressed in the matrix form predictor : $\underline{x}_{n,(\emptyset)} = B\underline{x}_{n-1}$ (3.3.5)

corrector : $\underline{x}_{n,(m+1)} = \underline{x}_{n,(m)} + \underline{c}F(\underline{x}_{n,(m)})$ (3.3.6)

where

$$B = \begin{bmatrix} 1 & \beta_1 & \beta_2 & \cdots & \beta_{p-1} & \beta_p \\ 0 & \delta_1 & \delta_2 & & \delta_{p-1} & \delta_p \\ 0 & 1 & 0 & & 0 & 0 \\ \vdots & & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & & 1 & 0 \end{bmatrix}$$

$$\underline{x}_n = \left[x_n, hx'_n, hx'_{n-1}, \cdots\cdots, hx'_{n-p+1} \right]^T, \ ' \equiv d/dt$$

$$\underline{x}_{n,(m)} = \left[x_{n,(m)}, hx'_{n,(m)}, hx'_{n-1,(m)}, \qquad hx'_{n-p+1,(m)} \right]^T$$

$$\underline{C} = \left[\beta^*_o, 1, \emptyset, \cdots\cdots, \emptyset \right]^T$$

$$F(\underline{x}_{n,(m)}) = hf(\underline{x}_{n,(m)}, t_n) - h\underline{x}'_{n,(m)}$$

where T is the transpose operator, $t_n = nh$,

and $hx'_{n,(m)} = hf(x_{n,(m-1)}, t_n)$ $\qquad m \geqslant 1$

$= h \sum_{i=1}^{p} \delta_i x'_{n-i}$ $\qquad m = \emptyset$

$$\delta_i = (\beta_i - \beta^*_i)/\beta^*_o, \quad x' \equiv dx/dt.$$

For stiff methods the equations are of the same form as (3.3.3) and (3.3.4) except that

$$\underline{x}_n = \left[x_n, hx'_n, x_{n-1}, \cdots\cdots, x_{n-p} \right]^T$$

$$B = \begin{bmatrix} \alpha_1 & \beta_1 & \alpha_2 & \cdots\cdots & \alpha_{p-1} & \alpha_p \\ \gamma_1 & \delta_1 & \gamma_2 & \cdots & \gamma_{p-1} & \gamma_p \\ 1 & 0 & 0 & \cdots\cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots\cdots & 0 & 0 \\ \vdots & & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \cdots\cdots & 1 & 0 \end{bmatrix}$$

$$\gamma_i = (\alpha_i - \alpha^*_i)/\eta^*_o$$

$$\delta_1 = \eta_1/\eta^*_o$$

and $\underline{c}$ is as before.

Rather than using the equations (3.3.3), (3.3.4) we use

$$\underline{z}_{n,\emptyset} = QBQ^{-1}\underline{z}_{n-1} \qquad (3.3.5)$$

$$\underline{z}_{n,(m+1)} = \underline{z}_{n,(m)} + IF(\underline{z}_{n,(m)}) \qquad (3.3.6)$$

where $I = Q\underline{c}$ and $\underline{z}_{n-1} = Q\underline{x}_{n-1}$.

The transformation Q is chosen so that the p + 1 components of $\underline{z}_{n-1}$ are the function value x_n and the first p derivatives of the polynomial used in the prediction process. The kth derivative is scaled by $h^k/k!$ so that the matrix Q is independent of h, and $\underline{z}_n$ becomes

$$\underline{z}_n = \left[x_n, hx_n', \ldots\ldots \; h^p x_n^{(p)}/p! \right]^T$$

where $x_n^{(k)}$ is the kth derivative of the approximating polynomial used by the predictor in the pth order method. QBQ^{-1} is the Pascal triangle matrix

$$\begin{bmatrix} 1 & 1 & 1 & 1 & \cdots & 1 & 1 \\ & 1 & 2 & 3 & \cdots & p-1 & p \\ & & 1 & 3 & & \vdots & \vdots \\ & & & 1 & & \vdots & \vdots \\ & \emptyset & & & \ddots & \vdots & \vdots \\ & & & & & 1 & p \\ & & & & & & 1 \end{bmatrix}$$

The back values are now those of a single point and this alleviates the problems usually arising in predictor-corrector methods when a change in step size results in the wrong back values being saved. The I for the Adam's and Stiff methods are given in [6], [7] respectively.

In the case of stiff equations, the corrector values will not converge unless h is very small. This results in either the solution process being slowed down dramatically or failing altogether. It has been found that using a modified form of corrector equation

$$\underline{z}_{n,(m+1)} = \underline{z}_{n,(m)} + IWF(\underline{z}_{n,(m)}) \qquad (3.3.7)$$

where $W = (-(\partial F/\partial \underline{z})I)^{-1}$

the iteration will converge to a solution of $\underline{F}(\underline{z}_n) = \emptyset$. For most functions $f(\underline{x},t)$ that occur, large values of h still permit rapid convergence. W is only re-evaluated if three iterations fails to bring the system close to convergence.

Initially a method of order one is chosen then $\underline{z}_\emptyset = (y_\emptyset, hy'_\emptyset)^T$, $y_\emptyset$ is given and $hy'_\emptyset$ can be calculated from $hf(y_\emptyset, t_\emptyset)$ so that the method is self starting. Changes of step size require only that $\underline{z}$ be scaled by powers of the change. Changes in order require reasonably easy modifications of $\underline{z}$.

3.4 Conclusions

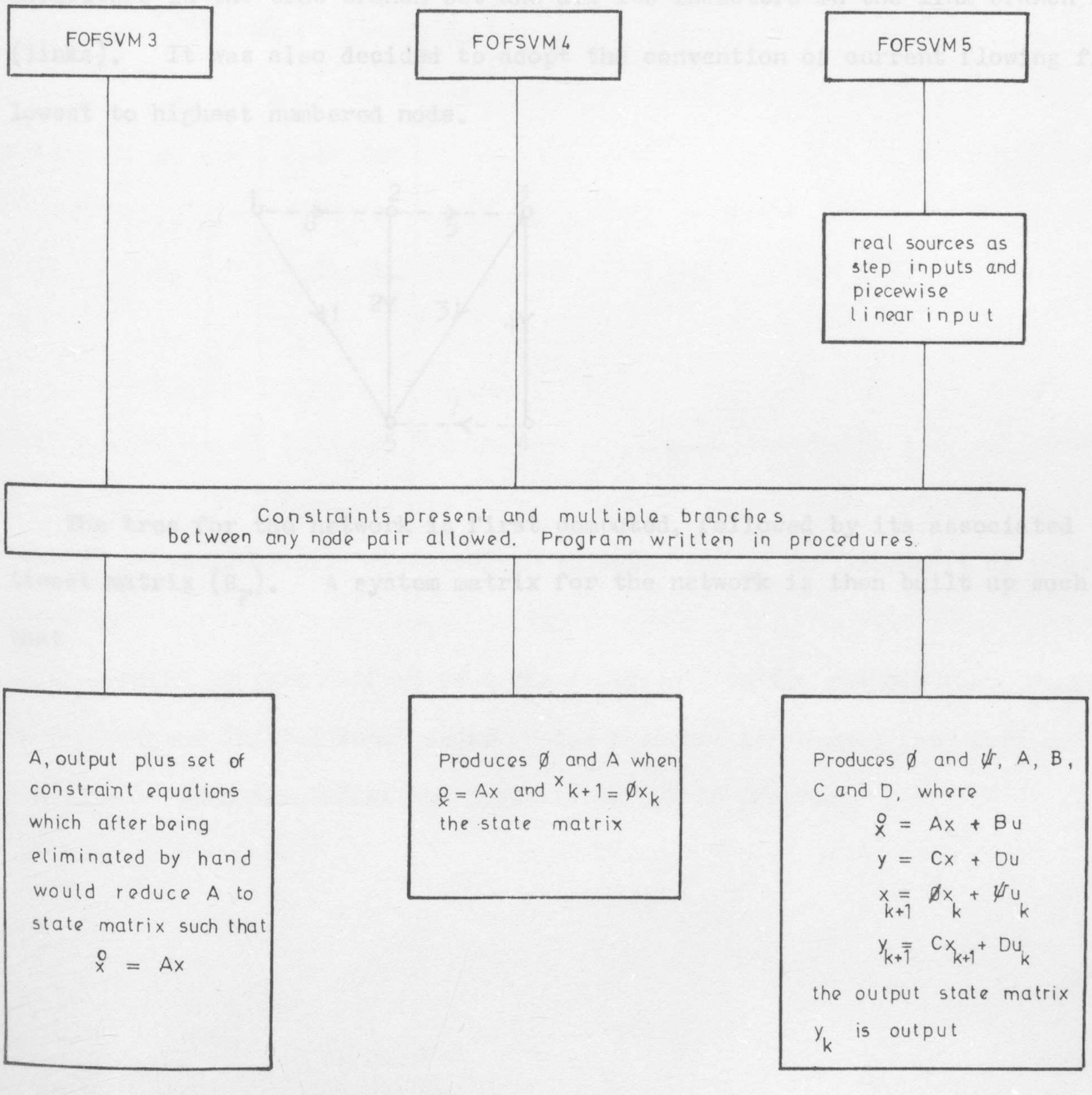
Network simplicity does not guarantee ease of numerical solution. The problems which may arise in the numerical solution are, in some cases readily predictable from the form of the original network, but, especially with large circuits unpredictable difficulties may arise.

A general purpose program therefore requires a powerful method of solution, to take account of this variable numerical complexity. The algorithm to ensure this, by Gear, has been chosen and described.

4. Development Programs

4.1 Introduction

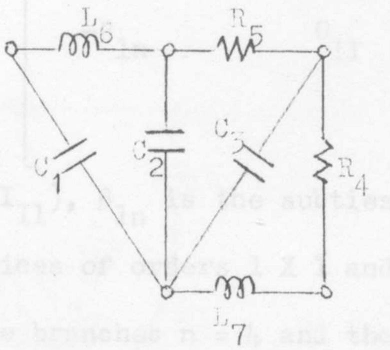
Six programs (FOFSVM1,2,3,4,5,6) have been developed which formulate the state equations for progressively more general networks using Bashkow's Topological Formulation. The last two programs also solve these equations. The earlier programs served to isolate and clarify the problems involved.


FOFSVM 3
FOFSVM4
FOFSVM5
real sources as step inputs and piecewise linear input
Constraints present and multiple branches between any node pair allowed. Program written in procedures.
A, output plus set of constraint equations which after being eliminated by hand would reduce A to state matrix such that
$\overset{o}{x} = Ax$
Produces Ø and A when $\overset{o}{x} = Ax$ and $x_{k+1} = Øx_k$ the state matrix
Produces Ø and ψ, A, B, C and D, where
$\overset{o}{x} = Ax + Bu$
$y = Cx + Du$
$x_{k+1} = Øx_k + \psi u_k$
$y_{k+1} = Cx_{k+1} + Du_k$
the output state matrix
y_k is output

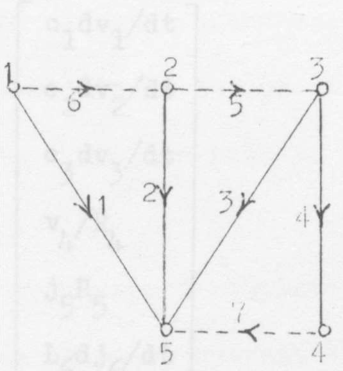
4.2 Programs FOFSVM1,2

The first and second programs were used to formulate the state equations for networks which contained only one branch between each node pair, each branch contained only one element, a resistor, a capacitor, or an inductor.

e.g.



A proper tree for the network could also be drawn, that is one which has all its capacitors in the tree branch set and all its inductors in the link branch set (links). It was also decided to adopt the convention of current flowing from lowest to highest numbered node.



The tree for the network is first computed, followed by its associated tieset matrix (β_F). A system matrix for the network is then built up such that

$$\begin{bmatrix} j_1 \\ j_2 \\ j_3 \\ j_4 \\ v_5 \\ v_6 \\ v_7 \end{bmatrix} = \begin{bmatrix} O_{nn} & \beta_{ln}^T \\ -\beta_{ln} & O_{ll} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ j_5 \\ j_6 \\ j_7 \end{bmatrix}$$

where $\beta_F = (\beta_{ln} \quad I_{ll})$, β_{ln} is the subtieset matrix and β_{ln}^T its transpose. O_{ll}, O_{nn} are null matrices of orders l X l and n X n respectively. In this case the number of tree branches $n = 4$ and the number of links $l = 3$. I_{ll} is a unit matrix of order l X l. Any equations representing resistive branches are now eliminated and the equations brought into the zero-input state form

$$d/dt \begin{bmatrix} \underline{v}_c \\ \underline{j}_L \end{bmatrix} = A \begin{bmatrix} \underline{v}_c \\ \underline{j}_L \end{bmatrix}$$

using

$$\begin{bmatrix} j_1 \\ j_2 \\ j_3 \\ j_4 \\ v_5 \\ v_6 \\ v_7 \end{bmatrix} = \begin{bmatrix} c_1 dv_1/dt \\ c_2 dv_2/dt \\ c_3 dv_3/dt \\ v_4/R_4 \\ j_5 R_5 \\ L_6 dj_6/dt \\ L_7 dj_7/dt \end{bmatrix}$$

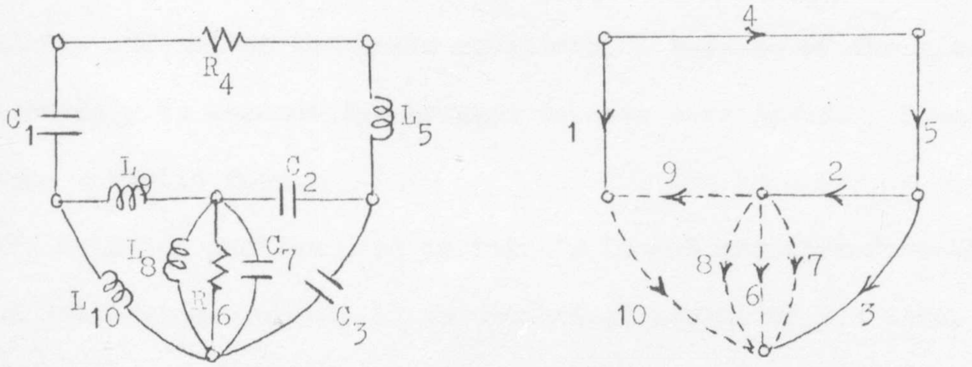
where $\underline{v}_c$ is the submatrix of capacitor voltages and $\underline{j}_L$ the submatrix of inductor currents.

The second program differs from the first only in the respect that, within the program the initial input order of the branches is changed into tree and link sets. This simplifies the logic later in the program.

4.3 Programs FOFSVM3,4

The third and fourth programs were used to formulate the state equations of more complicated networks. They could contain more than one branch between two nodes. Also loops of capacitors and cutsets of inductors were allowed, as were loops of inductors and cutsets of capacitors

e.g.



This time not all of the capacitor branches could be included in the tree branch set and not all of the inductors could be left out. A tree with the maximum number of capacitor carrying branches and a link set with the maximum number of inductor carrying branches is chosen. This is called a normal tree. An initial system matrix is set up as before, but now the constraints between the possible state variables have to be eliminated, as well as the resistor variables.

The third program cannot complete the formulation. It eliminates the resistor variables and also outputs the systems of constraint equations which have still to be solved and the results used to complete the formulation. The fourth program actually completes the formulation.

Both programs are written in procedure form. FOFSVM3 uses procedures called TREE, TIESET MATRIX, and ELEMENT SCAN. The latter procedure produces the equations of constraint. FOFSVM4 uses procedures TREE, TIESET MATRIX SUBSYSTEM, GFPIVOT and REDUCE MATRIX. SUBSYSTEM formulates the subsystems of equations of constraint. GFPIVOT is a Gaussian Full Pivoting procedure which solves the constraint subsystems. REDUCE MATRIX eliminates the dependent variables from the main system matrix. Two further procedures called TISETS and CUTSETS are used to count the numbers of tiesets and cutsets respectively. These numbers are used later on to decide if any further calls of other procedures are required.

4.4 Program FOFSVM5

This program is the same as FOFSVM4 except that it now accepts networks which contain independent sources, controlled sources and sinusoidal sources. All the previously mentioned procedures have been updated to deal with these sources. Also the program produces output-state variables selected by the user which can be on the lineprinter and on a graphical output device. The program formulates and solves the state equations. Because of the size of FOFSVM5 it was necessary to segment the program to save core space. Some procedures were put onto magnetic tape.

The solution section used Certaine's Method and therefore the program cannot solve some problems in an economical amount of run time. This is especially so with networks containing diodes and transistors which are represented by equivalent circuits by the user. In fact with a lot of the test problems of this type there have been troubles with relative stability. That is the transients output have been of the right shape, but grossly exaggerated in value.

The object of this program is to formulate and solve for quasi-linear and time-invariant networks (matrix coefficients A,B,C,D are constants, but sources may be non-linear) a pair of systems of equations

$$d\underline{x}(t)/dt = A\underline{x}(t) + B\underline{u}(x,t) \qquad (4.4.1)$$

$$\underline{y}(t) = c\underline{x}(t) + D\underline{u}(x,t) \qquad (4.4.2)$$

where (4.3.1) and (4.3.2) are the state and output-state equations respectively, $\underline{x}(t)$ is the state vector, $\underline{y}(t)$ the output-state vector, $\underline{u}(x,t)$ a vector of dependent and independent sources. In the formulation process the sources are treated as if they were all independent and are associated with a resistance, but during the solution the sources are updated as the computation progresses.

In the test problems tried with this program, failures occurred without exception when a row of the augmented matrix (A B) had all negative elements. This apparently causes spurious roots introduced by the numerical method to dominate the real roots, thereby introducing errors which are propagated exponentially, which give rise to exaggerated transients of the right shape. Another type of failure occurred when the solution process was progressing too slowly due to widely separated time constants and had to be abandoned.

4.5 Program FOFSVM6

This program uses matrix methods to produce a normal tree for a network. It uses two extra procedures INCIDENT MATRIX and COSORT. The incident matrix is first produced, then its columns are arranged by COSORT is some order of preference. This matrix is now operated on by GFPIVOT to bring it effectively into the form $(I_{nn} \; \alpha_{nl})$, where it is now identical to the cutset matrix α_F. The subcutset matrix α_{nl} is related to the subtieset matrix by the equation

$$\alpha_{nl} + \beta_{ln}^{T} = \underline{0}$$

T is the transpose operator, I_{nn} is a unit matrix of order n X n, l and n are the numbers of links and tree branches respectively. β_{ln} is produced and the analysis proceeds as in FOFSVM5. Procedures TREE and TIESET MATRIX are not used in FOFSVM6, their tasks being mainly done by GFPIVOT which results in about a saving of 1.5k of store. The program however needs further development before it reaches a satisfactory final form.

4.6 Conclusions

FOFSVM1,2,3,4 are no use as network analysis programs their function being developmental. FOFSVM5,6 can formulate the state equations for any RCL network which is time-invariant, and quasi-linear,

i.e. $d\underline{x}(t)/dt = A\underline{x}(t) + B\underline{u}(\underline{x}(t),t)$

where A and B are constant matrices. Any active devices such as diodes or transistors have to be replaced with equivalent circuit models by the user. The state variables chosen are always capacitor voltages and inductor currents. If any other variable is required as output it must be entered in the input data as an output-state variable. FOFSVM5,6 cannot solve any set of state equations which they formulate. Two types of failure can occur. The problem is relatively unstable for the numerical method of integration being used or the solution process is not moving fast enough as a large enough step size cannot be chosen without local errors becoming intolerable.

FOFSVM5 uses a search method of computing a normal tree for a network, whereas FOFSVM6 uses a matrix method. In both programs all sources are associated with a resistance. The method of Auxiliary Trees is used to formulate the equations for each type of constraint present in a network. The solution section uses a form of Certaine's Method with variable step-size routine.

5. Program Tests

5.Ø Introduction

FOFSVM5 is split into three segments only one of which is in core store at any time. The first is up to the formulation of a degenerate system of equations from which dependent variables and redundant equations need to be eliminated. The second segment reduces the degenerate system of equations to that of the State Variable Equations. The third solves the state variable equations and outputs data to the lineprinter and to a further program SVMPLT which produces graphs of the prescribed outputs.

5.1 Input of Data to FOFSVM5

The following list gives the format of data input for FOFSVM5.

```
'TITLE'

NRUN

B,RSOURCE,NY

┌ MN1(1)  MN2(1)  MCOMP(1)  ACOMP(1) ┐
│   :       :        :         :     │   branch identification
└ MN1(B)  MN2(B)  MCOMP(B)  ACOMP(B) ┘

┌ SMN1(1)         SMN2(1)         SMCOMP(1)         MSOURCE(1)        ┐
│   :                :               :                  :             │ source
└ SMN1(RSOURCE)   SMN2(RSOURCE)   SMCOMP(RSOURCE)   MSOURCE(RSOURCE)  ┘ identification

K1

A(Ø,1), A(Ø,2),.....,A(Ø,B)    initial values

YCOMP(1), YCOMP(2),.....,YCOMP(NY)   type of output

┌ NYBB(1,1),    NYBB(1,2),  .....  ,NYBB(1,2B)  ┐
│   :              :                    :       │   outputs required
└ NYBB(NY,1)    NYBB(NY,2),  .....  ,NYBB(NY,2B)┘

NINT(1),NINT(2),.....,NINT(ISOURCE)   number of time intervals for each
                                      independent source

┌ IS(1),THETA(1),AUX(1) ┐
│   :       :      :    │     sinusoidal sources
└                       ┘

┌ IS(1),THETA(1) ┐
│   :      :     │   diode sources
└                ┘

┌ IS(1) ┐
│   :   │   linear sources
└       ┘

┌ ((t_o,u_o),(t_1,u_1));((t_2,u_2),t_3,u_3));  ┐ piecewise linear continuous
│                :                             │ independent sources.
└                                              ┘

EPS,EPS1,EPS2,TØ,TN,NPTS,K2
```

The data is arranged ready for input in the above order though the presence of some items would depend on previous entries in the same run. For instance, entries in the integer array MSOURCE would indicate whether all or some of the permissable types of sources were present in the network under analysis.

'TITLE' name of network

NRUN run number set to a positive integer except for last run which is set to a negative integer.

B The total number of branches in the network excluding the sources which are always associated with a resistor.

RSOURCE The number of sources (which must be associated with resistors).

NY The number of outputs required.

MN1,MN2 Node pair (where the current direction is taken as that from the smallest to the largest of the pair).

MCOMP Element type identifier.

capacitor 1

resistor 2 entered by user

inductor 3

capacitor Ø<MCOMP<5Ø

resistor 5Ø<MCOMP<1ØØ within program

inductor 1ØØ<MCOMP<15Ø

ACOMP The numerical value of an element

SMN1,SMN2 Source node pair (written in direction of current or increasing voltage).

SMCOMP Type of source (voltage = 1, current = Ø)

MSOURCE Type of dependence of source:

Ø piecewise linear continuous function of time.

1 linearly controlled source

2 non-linearly controlled source (diode)

3 sinusoidal function of time.

A(Ø,1:B) Set of initial conditions (zero for resistors) which finally forms the initial state vector used in the solution process.

NYBB(1:NY,1:2B) Specification of NY required outputs (1 to B voltages and B + 1 to 2B currents).

EPS Truncation factor for transition matrix set to Ø.ØØØØØ1.

(TØ,TN) Time interval over which output is required.

K1 Integer indicator (Ø if all initial conditions zero 1 otherwise)

K2 Integer indicator (Ø if only output-state vector required on lineprinter 1 otherwise).

(EPS1,EPS2) Interval over which the relative change of a state variable from the K to K + 1 iteration is deemed tolerable.

$$\mathrm{EPS1} < \left| \frac{x(K+1)-x(K)}{x_{MAX}} \right| < \mathrm{EPS2}$$

YCOMP Type of output (current = Ø, voltage = 1).

NPTS Maximum number of points to be output after which the program goes onto the next run.

IS,THETA, AUX Real arrays representing parameters of the following types of sources:

sinusoidal: $u = i_s \sin(2\pi\Theta t + \varphi)$

where φ is the auxiliary angle.

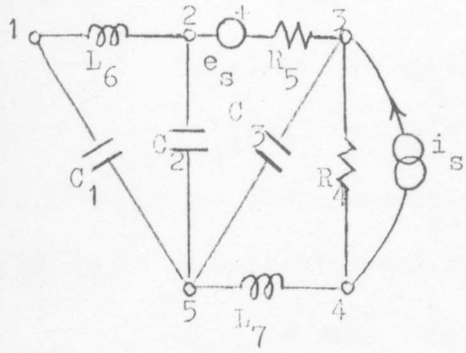
diode: $u = i_s(\exp(q/(\Theta kT))y(t)-1)$

where $y(t)$ is the controlling output-state variable.

linear: $u = i_s y(t)$.

$((t_o,u_o),(t_1,u_1))$ End points of a linear independent source over the time interval (t_o,t_1)

Example



´SIMPLE NETWORK WITH CONSTANT SOURCES`

NRUN = -1

B = 7, RSOURCE = 2, NY = 1

MNI	MNZ	MCOMP	ACOMP
1	2	1	C1
2	5	1	C2
3	5	1	C3
3	4	2	R4
2	3	2	R5
1	2	3	L6
4	5	3	L7

SMNI	SMNZ	SMCOMP	MSOURCE
4	3	Ø	Ø
2	3	1	Ø

K = Ø (all initial conditions to be taken as zero)

YCOMP = 1 (voltage output)

NYBB = 1,Ø,Ø,Ø,Ø,Ø,Ø; Ø,Ø,Ø,Ø,Ø,Ø,Ø

(capacitor voltage v_1 output)

NINT = 1,1 (one time interval for each source)

$((T\emptyset,i_s),(TN,i_s))$

$((T\emptyset,e_s),(TN,e_s))$ step inputs.

EPS,EPS1,EPS2,TØ,TN,NPTS = 5ØØ,

K = 1 (state variables to appear on lineprinter as well as output-state variables).

5.2 Output of Solution

The input data appears on the lineprinter, followed by the computed tree, the number of constraints, the order of the system of state equations, the matrices A,B,C,D where

$$\begin{aligned} d\underline{x}(t)/dt &= A\underline{x}(t) + B\underline{u}(t,\underline{x}(t)) \\ \underline{y}(t) &= C\underline{x}(t) + D\underline{u}(t,\underline{x}(t)) \end{aligned}$$

and finally the state and output-state vectors are output over the interval $T\emptyset \leqslant t \leqslant TN$ at variable step size H, via the difference equations

$$\underline{x}_{K+1} = \Phi\underline{x}_K + \Psi\underline{u}_K \text{ and } \underline{y}_{K+1} = C\underline{x}_{K+1} + D\underline{u}_K$$

where $\Phi = e^{AH}$ and $\Psi = (e^{AH}-I)A^{-1}B$ are the transition matrices produced within the program. The form of output of $\underline{x}(t)$ and $\underline{y}(t)$ would be

$$\begin{array}{lllllll} T = T\emptyset & x_1^{\emptyset} & x_2^{\emptyset} \cdots\cdots & x_{ORDER}^{\emptyset} & y_1^{\emptyset} & y_2^{\emptyset} \cdots\cdots & y_{NY}^{\emptyset} \\ \quad T + H & x_1^{1} & x_2^{1} \cdots\cdots & x_{ORDER}^{1} & y_1^{1} & y_2^{1} \cdots\cdots & y_{NY}^{1} \\ \quad \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots \\ \quad T+KH & x_1^{K} & x_2^{K} \cdots\cdots & x_{ORDER}^{K} & y_1^{K} & y_2^{K} \cdots\cdots & y_{NY}^{K} \end{array}$$

$\underline{y}(t)$ can also be output graphically via the program SVMPLT used in batch with FOFSVM5.

5.3 Data Pack

```
‘EXPONENTIAL  MODEL  OF  DIODE’
NRUN=2
B=6,RSOURCE=2,NY=2
2  1  1  2⏨-12
3  2  2  5
3  4  2  5
1  4  1  0.2⏨-9
1  4  2  10⏨3
2  1  2  100⏨3
1  2  0  2
3  4  1  0
K=0
YCOMP=1,1
V=1  0  0  0  0  0  J=0  0  0  0  0  0
V=1  1  0  0  0  0  J=0  0  0  0  0  0
NINT=2
IS=5⏨-11  THETA=1.5
0,-4,⏨-7,-4
⏨-7  +4  ⏨6  +4
EPS=⏨-6
EPSI=0.025  EPSZ=0.1
T0=0  TN=⏨-6  NPTS=500
K=1
‘NON-LINEAR MODEL  OF  TRANSISTOR’
NRUN=7
3=6  RSOURCE=2  NY=2
2  4  1  250⏨-12
3  4  1  100⏨-12
4  2  2  5⏨3
1  4  2  500
2  1  1  ⏨-7
4  3  2  5⏨3
3  4  0  2
4  1  1  0
K=0
YCOMP=1  1
V= 1  0  0  0  0  0  I=0  0  0  0  0  0
V=0  1  0  0  0  0  I=0  0  0  0  0  0
NINT=1
IS=5⏨-11  THETA=1.5
0  1  ⏨6  1
EPS=⏨-6
EPSI=0.025  EPS7=0.1
T0=0  TN=4.5⏨-6  NPTS=300
K=1
‘STEP  RESPONSE  OF  FILTER’
NRUN=4
```

```
B=9,RSOURCE=1,NY=1
1  5  1  279.860⏨-9
1  2  3  1.65407⏨-3
2  5  1  543.397⏨-9
2  3  3  39.7266⏨-3
3  5  1  936.027⏨-9
3  4  3  6.88075⏨-3
2  3  1  300.986⏨-9
1  5  2  100
4  5  2  100
5  1  1  0
K=0
YCOMP=0
V=0  0  0  0  0  0  0  0  0  J=0  0  0  0  0  1  0  0  0
NINT=1
0  1  ⏨6  1
EPS=⏨-6
EPSI=0.025  EPSZ=0.1
T0=0  TN=⏨30 NPTS=500
K:=0;
'BUTTERWORTH  FILTER'
NRUN=6
B=5,RSOURCE=1,NY=1
1  2  3  1
1  4.  2  1
2  3  3  1
2  4  1  2
3  4  2  1
4  1  1  0
K=0
YCOMP=0
V=0  0  0  0  0  J=0  0  1  0  0
NINT=2
0   100  1   100
1  0  ⏨6  0
EPS=⏨-6
EPSI=0.025  EPSZ=0.1
T0=0  TN=⏨30 NPTS=500
K:=0
'BRIDGED  TEE  FILTER'
NRUN=3
B=7,RSOURCE=1,NY=3
1  4  2  1000
2  4  1  ⏨-6
3  4  1  ⏨-6
1  3  1  ⏨-6
3  4  2  1000
1  2  3  1
```

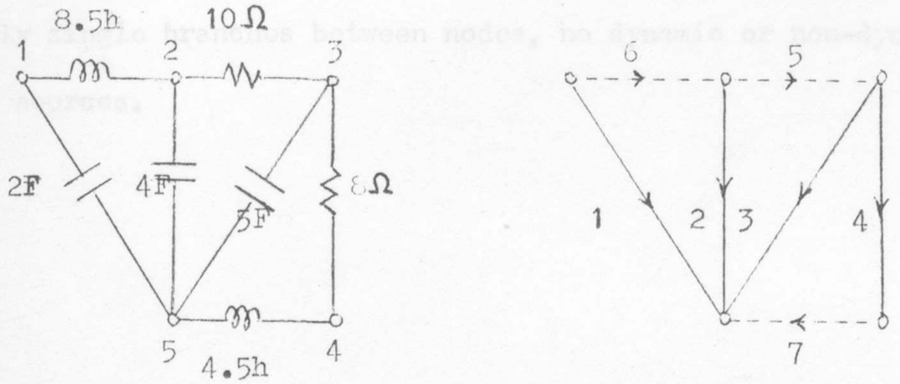
```
2  3    3  1.0
4  1  0  0
K=0
YCOMP=1,1,1
V=0  0  0  1  1  0  0  J=0  0  0  0  0  0  0
V=0  1  0  0  0  0  0  J=0  0  0  0  0  0  0
V=0  0  1  0  0  0  0  J=0  0  0  0  0  0  0
NINT=1
0   0.02  ₁₀6   0.02
EPS=₁₀-6
EPSI=0.025  EPSZ=0.1
TO=0  TN=₁₀30 NPTS=500
K=0
'LINEAR  MODEL  OF  TRANSISTOR`
NRUN=5
B=4  RSOURCE=2  NY=2
1  2  1  ₁₀-12
1  2  2  10
2  3  2  10
2  3  1  ₁₀-12
3  2  0  1
2  1  1  0
K=0
YCOMP=1  1
V=-1  0  0  0  I=0  0  0  0
V=0  0  0  -1  I=0  0  0  0
NINT=1
IS=0.5
0  1  ₁₀6  1
EPS=₁₀-6
EPSI=0.025  EPSZ=0.1
TO=0  TN=3₁₀-8  NPTS=500
K=1
'NETWORK  WITH  SINUSOIDAL  SOURCE`
NRUN=6
B=2,RSOURCE=1,NY=2
1  2  1  100₁₀-12
1  2  2  10₁₀3
2  1  1  3
K=0
YCOMP=1,1
V=0 -1  I=0  0
V=1  0  I=0  0
NINT=0
IS=1  THETA=5₁₀4  AUX=0
EPS=₁₀-6
EPSI=0.025  EPSZ=0.1
TO=0  TN=3₁₀-5  NPTS=300
```

```
K=1
'EXPONENTIAL MODEL OF TRANSISTOR'
NRUN=-1
B=5  RSOURCE=2  NY=3
1  3  1  500₁₀-12
1  3  2  1000
2  3  1  100₁₀-12
1  3  2  1000
2  3  2  1000
2  3  0  2
3  1  1  0
K=0
YCOMP=1  1  1
V=1  0  0  0  0  J=0  0  0  0  0
V=0  1  0  0  0  J=0  0  0  0  0
V=0  0  1  0  0  J=0  0  0  0  0
NINT=2
IS=₁₀-9  THETA=1
0  -1  ₁₀-6  -1
₁₀-6  1  ₁₀30  1
EPS=₁₀-6
EPSI=0.025  EPSZ=0.1
TO=0  TN=₁₀6  NPTS=500
K=1
```

5.4 Test Examples

The following five examples areformulation only. They have been computed using FOFSVM5.

Ex. 1



Referring to the computer printout overleaf, the number of branches B = 7, there are no sources RSCURCE = Ø, and there are no prescribed output state variables NY = Ø. The branch connecting nodes 1 and 5 (MN1(1) = 1, MN2(1) = 5) contains a capacitor (MCOMP(1) = 1) of 2 Farads (ACOMP(1) = 2). Similarly there are capacitive branches connecting nodes 2 and 5, and nodes 3 and 5. There are resistive branches (MCOMP = 2) connecting nodes 3 and 4, and nodes 2 and 3. Inductive branches (MCOMP = 3) connect nodes 1 and 2 and nodes 4 and 5. These branches can be input in any order between themselves but they must follow the first three numbers B, RSOURCE, NY.

The program computes the numbers of tree branches N = 4, links L = 3, resistors R = 2, capacitors C = 3, the order of the system of state equations representing the circuit, capacitor tiesets LC = Ø, resistor loops LC = Ø, inductor loops LL = Ø, capacitor cutsets NC = Ø, resistor cutsets NR = Ø, and inductor cutsets NL = Ø. This is followed by the subtieset matrix (β_{ln}), and its associated tree with branches TBl and links TLl relative to the order in which the branches were initially input.

In this case the smallest over largest pivot refers to the subsystem of resistor equations solved by procedure GFPIVOT and gives an indication of the numerical stability of the subsystem.

The state variables chosen by the program are v_1, v_2, v_3, j_6 and j_7 relative to the new order of the branches which in this case is identical to the original order. The state matrix A is computed where $d\underline{x}(t)/dt = A\underline{x}(t)$.

This was one of the simplest problems used to test the program. It has only single branches between nodes, no dynamic or non-dynamic constraints and no sources.

0

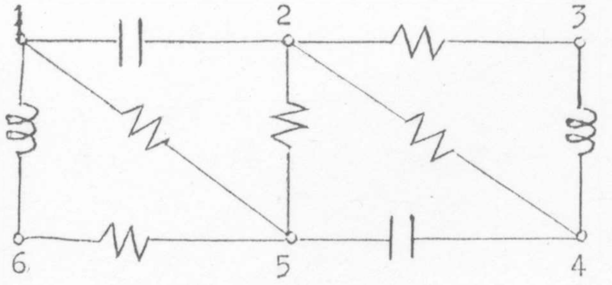
```
* * * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * * *
```

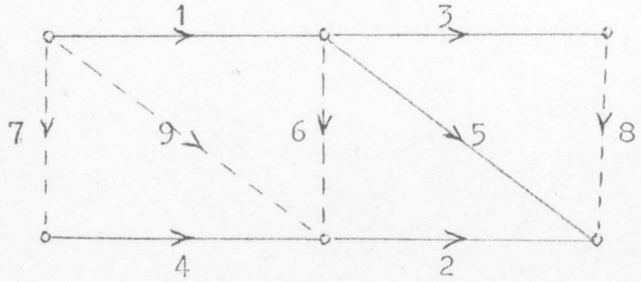
```
INPUT FOFSVM5;

B=    7  RSOURCE=     0  NY=     0
   B   MN1[B]  MN2[B]  MCOMP[B] ACOMP[B]
   1     1       5       1     2.000₁₀+00
   2     2       5       1     4.000₁₀+00
   3     3       5       1     5.000₁₀+00
   4     3       4       2     8.000₁₀+00
   5     2       3       2     1.000₁₀+01
   6     1       2       3     8.500₁₀+00
   7     4       5       3     4.500₁₀+00
N=    4   L=    3   R=    2   C=    3


OUTPUT FOFSVM5;
ORDER=    5   LC=    0   LR=    0   LL=    0   NC=    0   NR=    0   NL=

BETAF[L,N]=
    0   -1    1    0
   -1    1    0    0
    0    0   -1    1

OUTPUT  FOFSVM5;


TR1[N]=        1      2      3      4

TL1[L]=        5      6      7
NETWORK  BRANCHES  REORDERED  INTO  TREE  BRANCHES  AND  LINKS


        B  MN1[B] MN2[B] MCOMP[B] ACOMP[B]
        1      1      5      1    2.000₁₀+00    V[   1]
        2      2      5      2    4.000₁₀+00    V[   2]
        3      3      5      3    5.000₁₀+00    V[   3]
        4      3      4     51    8.000₁₀+00    V[   4]
        5      2      3     52    1.000₁₀+01    J[   5]
        6      1      2    101    8.500₁₀+00    J[   6]
        7      4      5    102    4.500₁₀+00    J[   7]

 SMALLEST OVER LARGEST PIVOT= 1.250₁₀-02
 THE  STATE  VARIABLES  ARE
 (   V[   1]   V[   2]   V[   3]   J[   6]   J[   7])

 A=
```

```
0.000₁₀+00     0.000₁₀+00     0.000₁₀+00    -5.000₁₀-01     0.000₁₀+00

0.000₁₀+00    -2.500₁₀-02     2.500₁₀-02     2.500₁₀-01     0.000₁₀+00

0.000₁₀+00     2.000₁₀-02    -2.000₁₀-02     0.000₁₀+00    -2.000₁₀-01

1.176₁₀-01    -1.176₁₀-01     0.000₁₀+00     0.000₁₀+00     0.000₁₀+00

0.000₁₀+00     0.000₁₀+00     2.222₁₀-01     0.000₁₀+00    -1.778₁₀+00
```

```
WHERE  DX/DT=AX
FORMULATION  ONLY
```

Ex. 2.





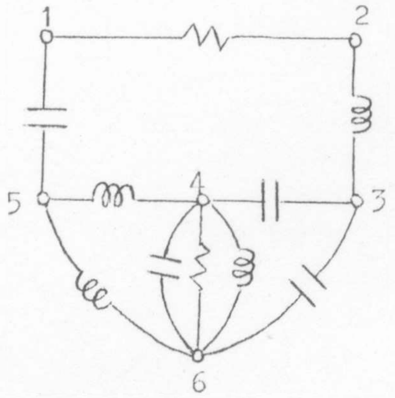
This circuit has a resistor current constraint (which would be called a cutset relative to a tree with the maximum number of resistor branches in the links). The problem tests that the resistor variables can be dealt with in similar manner to degenerate capacitor or inductor variables, by calls to procedures TREE, TIESET MATRIX, SUBSYSTEM, GFPIVOT and REDUCE MATRIX. The circuit still only contains single branches between nodes.

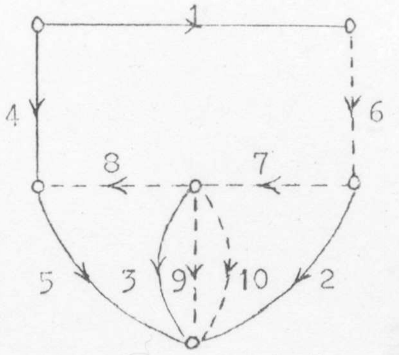
```
INPUT FOFSVM5;

B=     9   RSOURCE=     0   NY=     0
   B   MN1[B]  MN2[B]  MCOMP[B] ACOMP[B]

    1       1       2       1    2.000⏨+00
    2       4       5       1    2.500⏨+00
    3       2       3       2    4.000⏨+00
    4       2       5       2    5.000⏨+00
    5       5       6       2    6.000⏨+00
    6       1       6       3    7.000⏨+00
    7       3       4       3    8.000⏨+00
    8       1       5       2    8.500⏨+00
    9       2       4       2    9.000⏨+00
N=   5   L=   4   R=   5   C=   2


OUTPUT FOFSVM5;
ORDER=   4   LC=    0   LR=    0   LL=    0   NC=    0   NR=   1   NL=

BETAF[L,N]=
    0   -1    0    0   -1
   -1   -1    0   -1   -1
    0    0    1    0   -1
   -1   -1    0    0   -1

OUTPUT  FOFSVM5;


TR1[N]=      1      2      3      5      9

TL1[L]=      4      6      7      8
NETWORK  BRANCHES  REORDERED  INTO  TREE  BRANCHES  AND  LINKS


        B  MN1[B] MN2[B] MCOMP[B] ACOMP[B]
        1     1      2       1    2.000⏨+00    V[   1]
        2     4      5       2    2.500⏨+00    V[   2]
        3     2      3      51    4.000⏨+00    V[   3]
        4     5      6      53    6.000⏨+00    V[   4]
        5     2      4      55    9.000⏨+00    V[   5]
        6     2      5      52    5.000⏨+00    J[   6]
        7     1      6     101    7.000⏨+00    J[   7]
        8     3      4     102    8.000⏨+00    J[   8]
        9     1      5      54    8.500⏨+00    J[   9]
```

0

```
TB4=
      1       2       4       7       8
TL4=
      3       5       6       9

BETAF[L,N]=
     1     1     1    -1     1
     1     1     1    -1     0
     1     0     1    -1     0
     0     0     1    -1     0
SMALLEST OVER LARGEST PIVOT= 1.000₁₀+00
SMALLEST OVER LARGEST PIVOT= 1.961₁₀-02
THE  STATE  VARIABLES  ARE
(    V[   1]   V[   2]   J[   7]   J[   8])

A=
   -4.268₁₀-02   -1.524₁₀-02   -3.628₁₀-01    1.372₁₀-01

   -1.220₁₀-02   -3.293₁₀-02   -1.037₁₀-01    2.963₁₀-01

    1.037₁₀-01    3.702₁₀-02   -1.190₁₀+00   -3.332₁₀-01

   -3.430₁₀-02   -9.261₁₀-02   -2.915₁₀-01   -7.915₁₀-01




WHERE  DX/DT=AX
FORMULATION  ONLY
```

Ex. 3.



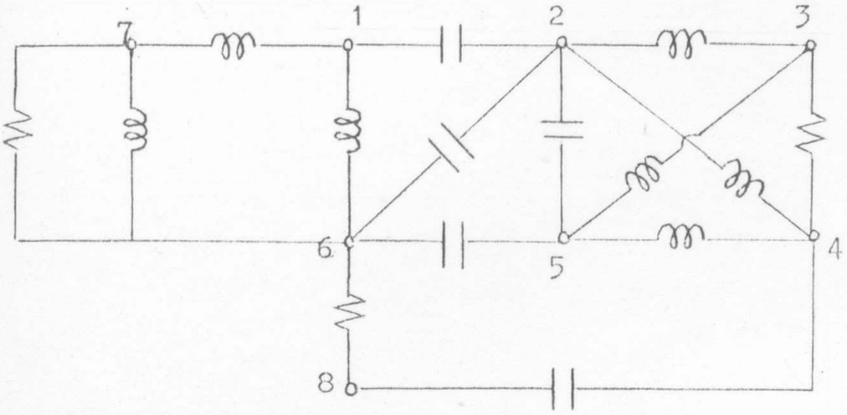


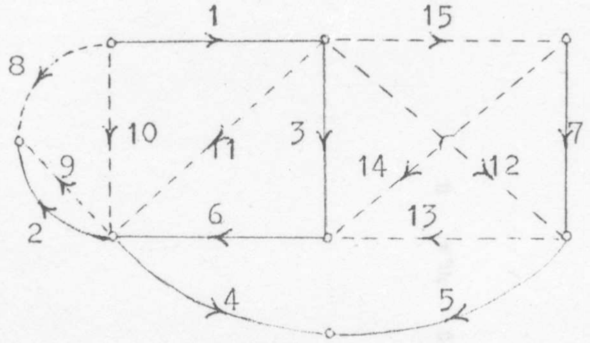
This circuit contains four dynamic constraints, one capacitor tieset, one inductor cutset, one loop of inductors (which would be called an inductor tieset relative to a tree with the maximum number of inductors in the tree branches) and one capacitor current constraint (which would be called a capacitor cutset relative to a tree with the maximum number of capacitors in the links). This tests the sequence of procedure calls which have to be made to eliminate the different types of constraint subsystems which are interactive with each other and the main system of equations. It also overlaps with the first problem as it has to eliminate a resistor subsystem from the main set of equations. This problem also contains three branches between one pair of nodes.

```
INPUT FOFSVM5;

B=  10   RSOURCE=     0   NY=     0
   B   MN1[B]  MN2[B]  MCOMP[B] ACOMP[B]

   1      1       2       2    5.000₁₀+00
   2      2       3       3    2.000₁₀+00
   3      3       4       1    3.000₁₀+00
   4      3       6       1    6.000₁₀+00
   5      4       5       3    7.000₁₀+00
   6      4       6       1    9.000₁₀+00
   7      4       6       2    4.000₁₀+00
   8      4       6       3    2.500₁₀+00
   9      1       5       1    8.000₁₀+00
  10      5       6       3    7.500₁₀+00
N=   5   L=    5   R=    2   C=   4


OUTPUT FOFSVM5;
ORDER=   4   LC=   1   LR=    0   LL=    1   NC=    1   NR=    0   NL=    1

BETAF[L,N]=
     1    1    0   -1   -1
     0   -1    1    0    0
     0    0   -1    0    1
     0    0   -1    0    0
     0    0   -1    0    0

OUTPUT  FOFSVM5;


TR1[N]=        1       4       6       9      10

TL1[L]=        2       3       5       7       8
NETWORK  BRANCHES  REORDERED  INTO  TREE  BRANCHES  AND  LINKS


        B   MN1[B] MN2[B] MCOMP[B] ACOMP[B]
        1       1       2      51    5.000₁₀+00    V[    1]
        2       3       6       2    6.000₁₀+00    V[    2]
        3       4       6       3    9.000₁₀+00    V[    3]
        4       1       5       4    8.000₁₀+00    V[    4]
        5       5       6     104    7.500₁₀+00    V[    5]
        6       2       3     101    2.000₁₀+00    J[    6]
        7       3       4       1    3.000₁₀+00    J[    7]
        8       4       5     102    7.000₁₀+00    J[    8]
```

```
      9        4        6       52    4.000⏨+00    J[    9]
     10        4        6      103    2.500⏨+00    J[   10]

TR2=
      1        5        6        7      10
TL2=
      2        3        4        8       9

BETAF[L,N]=
     0      0      0     -1     -1
     0      0      0      0     -1
    -1      1     -1     -1     -1
     0      1      0      0     -1
     0      0      0      0     -1
SMALLEST OVER LARGEST PIVOT= 1.000⏨+00
SMALLEST OVER LARGEST PIVOT= 1.000⏨+00
SMALLEST OVER LARGEST PIVOT= 1.000⏨+00
SMALLEST OVER LARGEST PIVOT= 1.000⏨+00
SMALLEST OVER LARGEST PIVOT= 5.000⏨-02
THE  STATE  VARIABLES  ARE
(    V[   3]   V[    4]   J[    6]   J[  10])

A=
    -2.273⏨-02     0.000⏨+00    -1.672⏨-02    -1.066⏨-01

     0.000⏨+00     0.000⏨+00    -1.250⏨-01     0.000⏨+00

     3.272⏨-02     3.361⏨-01    -8.896⏨-01     0.000⏨+00

     4.000⏨-01     0.000⏨+00     0.000⏨+00     0.000⏨+00







WHERE  DX/DT=AX
FORMULATION  ONLY
```

Ex. 4.





In this case, the network contains one capacitor tieset, two inductor loops, and one capacitor current constraint. Procedure GFPIVOT operates on one rectangular dynamic constraint subsystem of order 2 X 7 and on a resistor subsystem of order 3 X 15. This tests further the method of elimination of constraints and resistor variables.

```
INPUT FOFSVM5;

B=   15   RSOURCE=   0   NY=   0
   B   MN1[B]  MN2[B]  MCOMP[B] ACOMP[B]

   1      1      7       3    1.500₁₀+00
   2      1      2       1    2.000₁₀+00
   3      6      7       2    2.500₁₀+00
   4      6      7       3    3.000₁₀+00
   5      1      6       3    3.500₁₀+00
   6      2      6       1    4.500₁₀+00
   7      2      5       1    5.000₁₀+00
   8      6      8       2    5.500₁₀+00
   9      4      8       1    6.000₁₀+00
  10      5      6       1    6.500₁₀+00
  11      2      4       3    7.000₁₀+00
  12      4      5       3    7.500₁₀+00
  13      3      4       2    8.000₁₀+00
  14      3      5       3    8.500₁₀+00
  15      2      3       3    9.000₁₀+00
N=   7   L=   8   R=   3   C=   5


OUTPUT FOFSVM5;
ORDER=   8   LC=   1   LR=   0   LL=   2   NC=   1   NR=   0   NL=

BETAF[L,N]=
   -1   -1   -1    0    0   -1    0
    0   -1    0    0    0    0    0
   -1    0   -1    0    0   -1    0
    0    0   -1    0    0   -1    0
    0    0   -1   -1    1   -1    0
    0    0    0    1   -1    1    0
    0    0    0    1   -1    1   -1
    0    0   -1   -1    1   -1    1

OUTPUT  FOFSVM5;


TR1[N]=     2      3      7      8      9     10     13

TL1[L]=     1      4      5      6     11     12     14     15
NETWORK  BRANCHES  REORDERED  INTO  TREE  BRANCHES  AND  LINKS


     B  MN1[B] MN2[B] MCOMP[B] ACOMP[B]
```

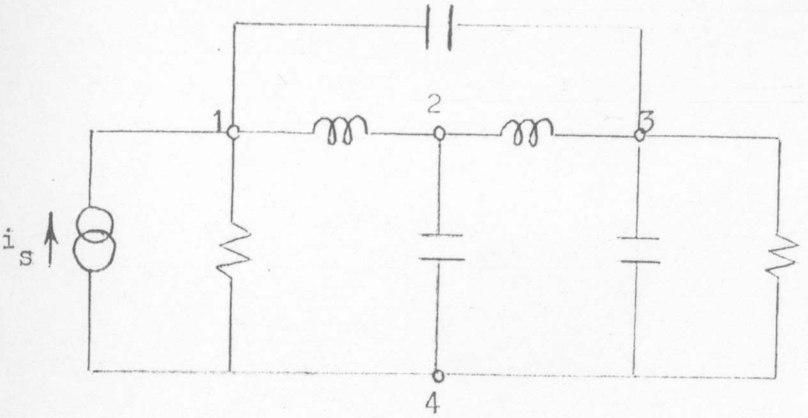
```
      1       1       2       1     2.000⏨+00    V[   1]
      2       6       7      51     2.500⏨+00    V[   2]
      3       2       5       3     5.000⏨+00    V[   3]
      4       6       8      52     5.500⏨+00    V[   4]
      5       4       8       4     6.000⏨+00    V[   5]
      6       5       6       5     6.500⏨+00    V[   6]
      7       3       4      53     8.000⏨+00    V[   7]
      8       1       7     101     1.500⏨+00    J[   8]
      9       6       7     102     3.000⏨+00    J[   9]
     10       1       6     103     3.500⏨+00    J[  10]
     11       2       6       2     4.500⏨+00    J[  11]
     12       2       4     104     7.000⏨+00    J[  12]
     13       4       5     105     7.500⏨+00    J[  13]
     14       3       5     106     8.500⏨+00    J[  14]
     15       2       3     107     9.000⏨+00    J[  15]

TR2=
      4       6       9      10       13      14      15
TL2=
      1       2       3       5        7       8      11      12

BETAF[L,N]=
     0     1     0    -1     0     1     1
     0     0    -1     0     0     0     0
     0     0     0     0     0    -1    -1
    -1    -1     0     0    -1     0     0
     0     0     0     0     1    -1     0
     0     0    -1    -1     0     0     0
     0    -1     0     0     0    -1    -1
     0     0     0     0     1    -1    -1
SMALLEST OVER LARGEST PIVOT= 1.000⏨+00
SMALLEST OVER LARGEST PIVOT= 1.000⏨+00
SMALLEST OVER LARGEST PIVOT= 3.889⏨-01
SMALLEST OVER LARGEST PIVOT= 3.125⏨-01
THE  STATE  VARIABLES  ARE
(   V[   1]   V[   3]   V[   5]   J[   8]   J[   9]   J[  12]   J[

A=
    0.000⏨+00     0.000⏨+00     0.000⏨+00    -7.143⏨-01     4.286⏨-01

    0.000⏨+00     0.000⏨+00     0.000⏨+00    -1.102⏨-01     6.613⏨-02

    0.000⏨+00     0.000⏨+00     0.000⏨+00     0.000⏨+00     0.000⏨+00

    7.879⏨-01     3.939⏨-01    -3.636⏨-01    -1.667⏨+00    -1.667⏨+00

    0.000⏨+00     0.000⏨+00     0.000⏨+00    -8.333⏨-01    -8.333⏨-01
13]    J[   14])

 0.000⏨+00      0.000⏨+00      0.000⏨+00

-1.372⏨-01     -1.177⏨-01      1.945⏨-02

 2.963⏨-01     -2.778⏨-02     -3.241⏨-01

 0.000⏨+00      0.000⏨+00      0.000⏨+00

 0.000⏨+00      0.000⏨+00      0.000⏨+00
```

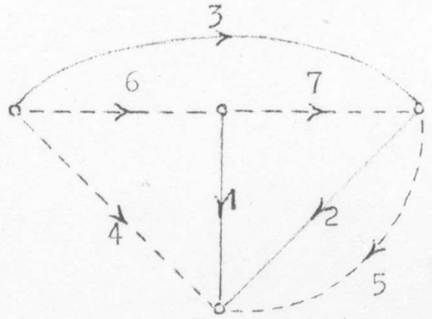
```
  2.597⏨-02    8.442⏨-02   -2.208⏨-01    0.000⏨+00    0.000⏨+00

 -2.424⏨-02    5.455⏨-02    2.061⏨-01    0.000⏨+00    0.000⏨+00

 -2.139⏨-02    4.813⏨-02    1.818⏨-01    0.000⏨+00    0.000⏨+00
```

```
WHERE  DX/DT=AX
FORMULATION  ONLY
```

```
-1.397⏨+00     1.310⏨-01     1.528⏨+00

 1.304⏨+00    -1.222⏨-01    -1.426⏨+00

 1.882⏨+00     6.765⏨-01    -3.088⏨+00
```

Ex. 5.





This problem has an independent current source and prescribed output state variables v_4, v_1 and v_2, so that as well as the state matrix A, extra matrices B, C and D have to be computed; where B is the source matrix in the state equations, C and D are the output state matrices, such that

$$d\underline{x}(t)/dt = A\underline{x}(t) + B\underline{u}(t)$$

and

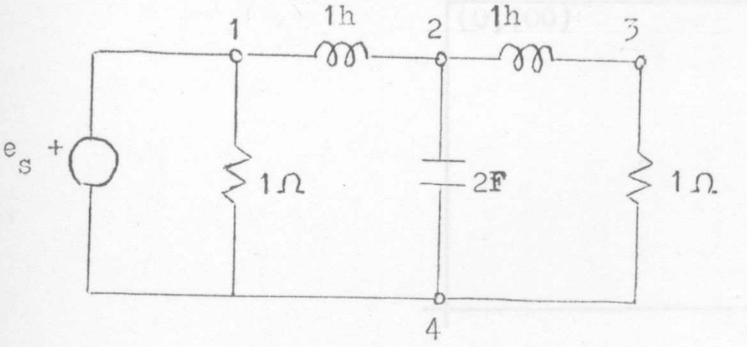
$$\underline{y}(t) = C\underline{x}(t) + D\underline{u}(t).$$

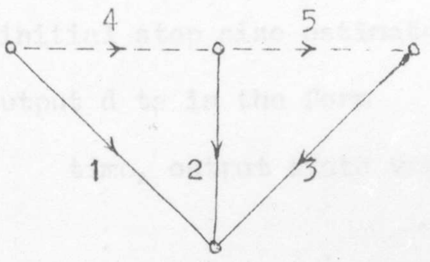
```
INPUT FOFSVM5;

B=    7   RSOURCE=     1   NY=    3
   B   MN1[B]  MN2[B]  MCOMP[B] ACOMP[B]

   1     1       4       2     1.000⏨+03
   2     2       4       1     1.000⏨-06
   3     3       4       1     1.000⏨-06
   4     1       3       1     1.000⏨-06
   5     3       4       2     1.000⏨+03
   6     1       2       3     1.000⏨+00
   7     2       3       3     1.000⏨+00
     SMN1    SMN2  SMCOMP    SCOL   MSOURCE
        4       1       0       1        0

N=    3    L=    4   R=    2    C=    3


OUTPUT FOFSVM5;
ORDER=    5    LC=    0    LR=    0    LL=    0    NC=    0    NR=    0    NL=    0

BETAF[L,N]=
    0    -1    -1
    0    -1     0
    1    -1    -1
   -1     1     0

OUTPUT  FOFSVM5;


TB1[N]=        2       3       4

TL1[L]=        1       5       6       7
NETWORK  BRANCHES  REORDERED  INTO  TREE  BRANCHES  AND  LINKS


        B  MN1[B] MN2[B] MCOMP[B] ACOMP[B]
        1      2      4        1   1.000⏨-06   V[    1]
        2      3      4        2   1.000⏨-06   V[    2]
        3      1      3        3   1.000⏨-06   V[    3]
        4      1      4       51   1.000⏨+03   J[    4]
        5      3      4       52   1.000⏨+03   J[    5]
        6      1      2      101   1.000⏨+00   J[    6]
        7      2      3      102   1.000⏨+00   J[    7]

     SMN1  SMN2  SMCOMP  SCOL  MSOURCE
```

```
         4          1          0          4          0

     SMALLEST OVER LARGEST PIVOT= 1.000₁₀+00
     THE  STATE  VARIABLES  ARE
     (   V[   1]   V[   2]   V[   3]   J[   6]   J[   7])

     A=
          0.000₁₀+00    0.000₁₀+00    0.000₁₀+00    1.000₁₀+06

          0.000₁₀+00   -2.000₁₀+03   -1.000₁₀+03   -1.000₁₀+06

          0.000₁₀+00   -1.000₁₀+03   -1.000₁₀+03   -1.000₁₀+06

         -1.000₁₀+00    1.000₁₀+00    1.000₁₀+00    0.000₁₀+00

          1.000₁₀+00   -1.000₁₀+00    0.000₁₀+00    0.000₁₀+00


     B=
          0.000₁₀+00

          1.000₁₀+06

          1.000₁₀+06

          0.000₁₀+00

          0.000₁₀+00


     C=
          0.000₁₀+00    1.000₁₀+00    1.000₁₀+00    0.000₁₀+00

          1.000₁₀+00    0.000₁₀+00    0.000₁₀+00    0.000₁₀+00

          0.000₁₀+00    1.000₁₀+00    0.000₁₀+00    0.000₁₀+00


     D=
          0.000₁₀+00

          0.000₁₀+00

          0.000₁₀+00


     WHERE  DX/DT=AX+BU    AND   Y=CX+DU
```

```
-1.000$_{10}$+06

 1.000$_{10}$+06

 0.000$_{10}$+00

 0.000$_{10}$+00

 0.000$_{10}$+00







 0.000$_{10}$+00

 0.000$_{10}$+00

 0.000$_{10}$+00
```

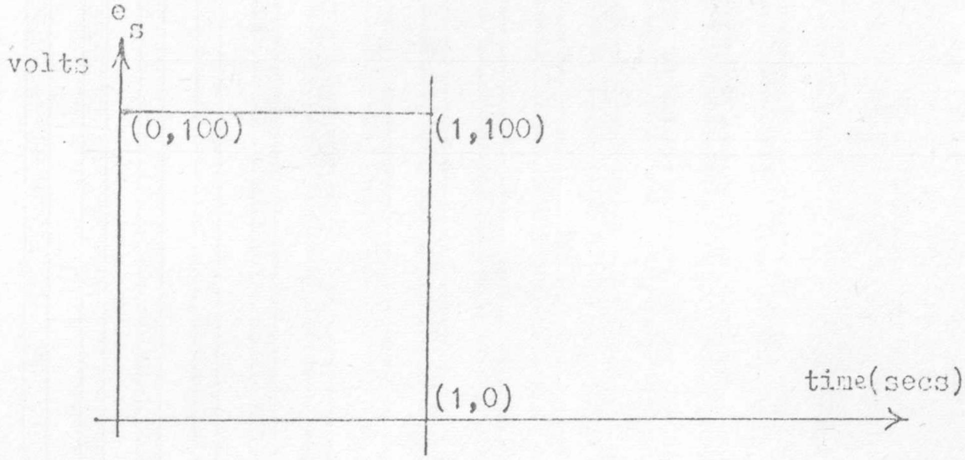
Ex. 6. Butterworth Filter





This is the first of a series of problems for which the state equations for networks are formulated and solved. A voltage source of 100 volts is switched on for one second and the transient current response j_5 through the second inductor is observed. The current carries on rising after the source has been switched off, peaks when the energy limit is reached and then tails off with some oscillation to steady off back at zero after about twenty seconds have elapsed, since the initial voltage input.

The computer printout now contains the source input data. There is an independent (MSOURCE(1) = Ø) voltage source (SMCOMP(1) = 1) between nodes 4 and 1, i.e. the voltage rise is from node 4 to node 1. More input data referring to the source voltage u was required. There are two piecewise time intervals NINT(1) = 2, and two sets of coordinates (TINT,UINT) of the end points of the graphs of the independent source.



The initial step size estimate H computed by the program is used to produce output data in the form

```
time, output state variables
 .                 .
 .                 .
 .                 .
 .                 .
```

The output of state variables to the lineprinter was suppressed in this example by setting K = Ø at the end of the list of input data. VMAX and IMAX are used to monitor local errors during the solution process. The output state variable data is also output on an off-line graph plotter via the program SVMPLT.

Referring to the plotter output of inductor current versus time, p65 (vi), superimposed points are from p.112, reference 11. The output from FOFSVM5 corresponds very favourably, the greatest discrepancy being at the peak overshoot.

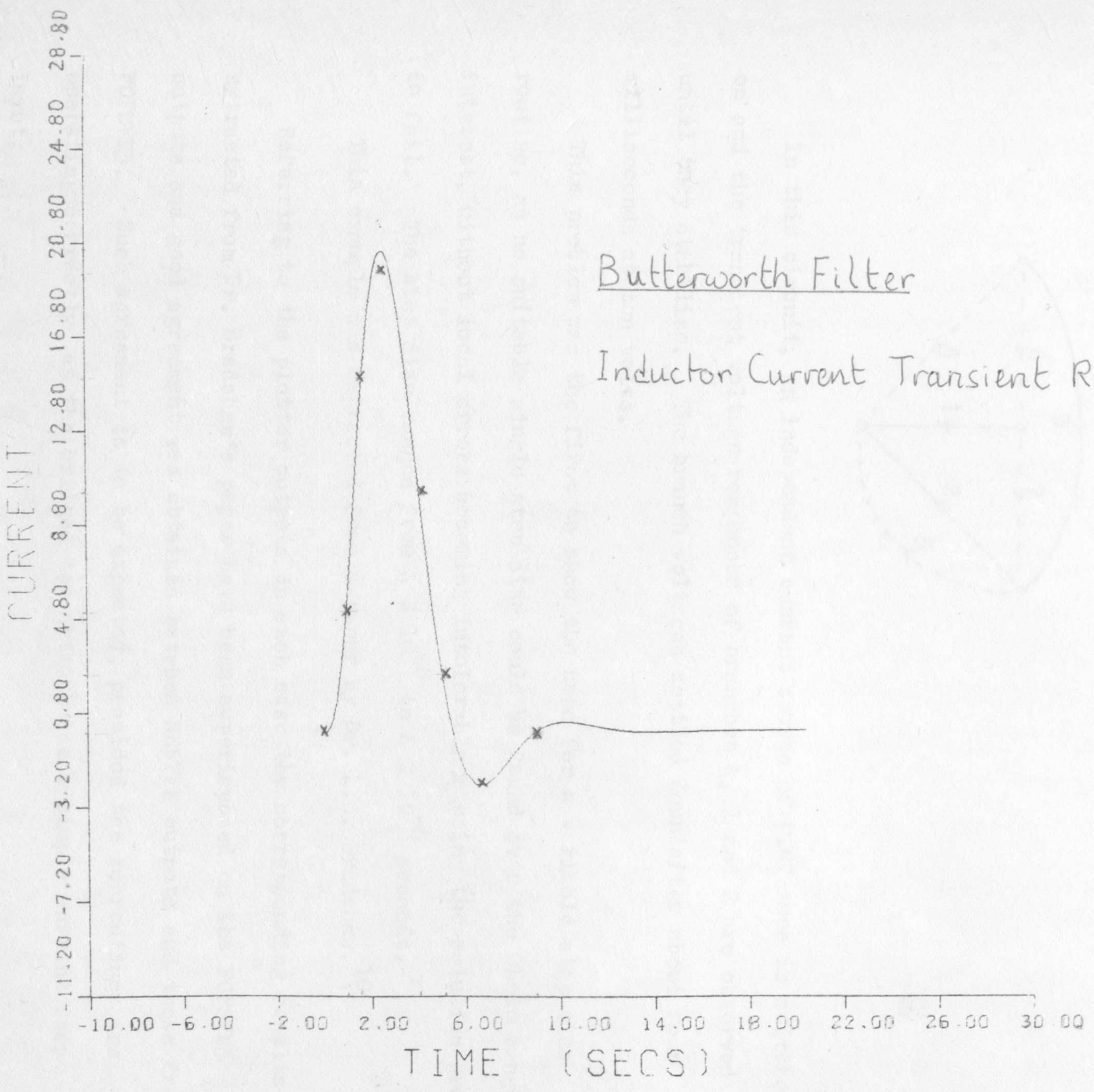
```
BUTTERWORTH  FILTER
FOFSVM5 OUTPUT
NRUN=   6

INPUT FOFSVM5:

B=     5   RSOURCE=     1   NY=     1
     B   MN1[B]  MN2[B]  MCOMP[B] ACOMP[B]

     1        1        2        3    1.000⏨+00
     2        1        4        2    1.000⏨+00
     3        2        3        3    1.000⏨+00
     4        2        4        1    2.000⏨+00
     5        3        4        2    1.000⏨+00
       SMN1    SMN2   SMCOMP    SCOL   MSOURCE
          4         1        1        2        0

N=    3   L=    2   R=    2   C=    1


OUTPUT FOFSVM5;
ORDER=   3    LC=    0    LR=    0    LL=    0    NC=    0    NR=    0    NL=    0

BETAF[L,N]=
    -1     1     0
     0    -1     1

OUTPUT   FOFSVM5;


TB1[N]=         2        4        5

TL1[L]=         1        3
NETWORK  BRANCHES  REORDERED  INTO  TREE  BRANCHES  AND  LINKS


         B   MN1[B] MN2[B] MCOMP[B] ACOMP[B]
         1      1       4      51     1.000⏨+00    V[    1]
         2      2       4       1     2.000⏨+00    V[    2]
         3      3       4      52     1.000⏨+00    V[    3]
         4      1       2     101     1.000⏨+00    J[    4]
         5      2       3     102     1.000⏨+00    J[    5]

       SMN1  SMN2  SMCOMP  SCOL  MSOURCE
          4       1       1       1       0

SMALLEST OVER LARGEST PIVOT= 1.000⏨+00
THE  STATE  VARIABLES  ARE
```

```
(    V[    2]   J[    4]    J[    5])

A=
      0.000⏨+00      5.000⏨-01     -5.000⏨-01

     -1.000⏨+00     -1.000⏨+00      0.000⏨+00

      1.000⏨+00      0.000⏨+00     -1.000⏨+00


B=
      0.000⏨+00

      1.000⏨+00

      0.000⏨+00


C=
      0.000⏨+00      0.000⏨+00      1.000⏨+00


D=
      0.000⏨+00



WHERE  DX/DT=AX+BU   AND  Y=CX+DU

H= 4.000⏨-02
NINT=        2
TINT= 0.000⏨+00   UINT= 1.000⏨+02
TINT= 1.000⏨+00   UINT= 1.000⏨+02
TINT= 1.000⏨+00   UINT= 0.000⏨+00
TINT= 1.000⏨+06   UINT= 0.000⏨+00
EPS= 1.000⏨-06  EPS1= 2.500⏨-02  EPS2= 1.000⏨-01
T0= 0.000⏨+00  TN= 1.000⏨+30  NPTS=        500

  F=0  T= 0.000⏨+00       0.000⏨+00
VMAX= 1.000⏨+02    IMAX= 1.000⏨+02

        1     4.000⏨-02     5.228⏨-04
        2     8.000⏨-02     4.099⏨-03
        3     1.200⏨-01     1.356⏨-02
        4     1.600⏨-01     3.149⏨-02
        5     2.000⏨-01     6.026⏨-02
        6     2.400⏨-01     1.020⏨-01
        7     2.800⏨-01     1.587⏨-01
```

```
       8     3.200⏨-01      2.321⏨-01
       9     3.600⏨-01      3.237⏨-01
      10     4.000⏨-01      4.349⏨-01
      11     4.400⏨-01      5.669⏨-01
      12     4.800⏨-01      7.208⏨-01
      13     5.200⏨-01      8.974⏨-01
      14     5.600⏨-01      1.097⏨+00
      15     6.000⏨-01      1.322⏨+00
      16     6.400⏨-01      1.570⏨+00
      17     6.800⏨-01      1.844⏨+00
      18     7.200⏨-01      2.142⏨+00
      19     7.600⏨-01      2.466⏨+00
      20     8.000⏨-01      2.815⏨+00
 STEP SIZE INCREASED        H= 8.000⏨-02

      21     8.800⏨-01      3.588⏨+00
      22     9.600⏨-01      4.459⏨+00
      23     1.040⏨+00      5.426⏨+00
      24     1.120⏨+00      6.479⏨+00
      25     1.200⏨+00      7.595⏨+00
      26     1.280⏨+00      8.748⏨+00
      27     1.360⏨+00      9.915⏨+00
      28     1.440⏨+00      1.108⏨+01
      29     1.520⏨+00      1.221⏨+01
      30     1.600⏨+00      1.331⏨+01
      31     1.680⏨+00      1.435⏨+01
      32     1.760⏨+00      1.533⏨+01
      33     1.840⏨+00      1.624⏨+01
      34     1.920⏨+00      1.707⏨+01
      35     2.000⏨+00      1.781⏨+01
      36     2.080⏨+00      1.846⏨+01
      37     2.160⏨+00      1.902⏨+01
      38     2.240⏨+00      1.948⏨+01
      39     2.320⏨+00      1.985⏨+01
      40     2.400⏨+00      2.012⏨+01
      41     2.480⏨+00      2.030⏨+01
      42     2.560⏨+00      2.039⏨+01
      43     2.640⏨+00      2.038⏨+01
      44     2.720⏨+00      2.030⏨+01
      45     2.800⏨+00      2.013⏨+01
      46     2.880⏨+00      1.989⏨+01
 STEP SIZE INCREASED        H= 2.000⏨-01

      47     3.080⏨+00      1.897⏨+01
      48     3.280⏨+00      1.769⏨+01
      49     3.480⏨+00      1.613⏨+01
      50     3.680⏨+00      1.436⏨+01
      51     3.880⏨+00      1.249⏨+01
```

```
      52     4.080₁₀+00      1.056₁₀+01
      53     4.280₁₀+00      8.667₁₀+00
      54     4.480₁₀+00      6.845₁₀+00
      55     4.680₁₀+00      5.144₁₀+00
      56     4.880₁₀+00      3.596₁₀+00
      57     5.080₁₀+00      2.223₁₀+00
      58     5.280₁₀+00      1.040₁₀+00
      59     5.480₁₀+00      5.057₁₀-02
      60     5.680₁₀+00     -7.477₁₀-01
  STEP SIZE INCREASED    H= 4.000₁₀-01

      61     6.080₁₀+00     -1.808₁₀+00
      62     6.480₁₀+00     -2.256₁₀+00
      63     6.880₁₀+00     -2.245₁₀+00
      64     7.280₁₀+00     -1.933₁₀+00
      65     7.680₁₀+00     -1.464₁₀+00
      66     8.080₁₀+00     -9.539₁₀-01
      67     8.480₁₀+00     -4.840₁₀-01
      68     8.880₁₀+00     -1.035₁₀-01
      69     9.280₁₀+00      1.666₁₀-01
      70     9.680₁₀+00      3.271₁₀-01
  STEP SIZE INCREASED    H= 8.000₁₀-01

  STEP SIZE DECREASED HMAX= 4.000₁₀-01

      71     1.008₁₀+01      3.929₁₀-01
      72     1.048₁₀+01      3.864₁₀-01
      73     1.088₁₀+01      3.320₁₀-01
      74     1.128₁₀+01      2.526₁₀-01
      75     1.168₁₀+01      1.667₁₀-01
      76     1.208₁₀+01      8.749₁₀-02
      77     1.248₁₀+01      2.308₁₀-02
      78     1.288₁₀+01     -2.305₁₀-02
      79     1.328₁₀+01     -5.094₁₀-02
      80     1.368₁₀+01     -6.299₁₀-02
      81     1.408₁₀+01     -6.286₁₀-02
      82     1.448₁₀+01     -5.458₁₀-02
      83     1.488₁₀+01     -4.192₁₀-02
      84     1.528₁₀+01     -2.798₁₀-02
      85     1.568₁₀+01     -1.499₁₀-02
      86     1.608₁₀+01     -4.327₁₀-03
      87     1.648₁₀+01      3.383₁₀-03
      88     1.688₁₀+01      8.111₁₀-03
      89     1.728₁₀+01      1.023₁₀-02
      90     1.768₁₀+01      1.031₁₀-02
      91     1.808₁₀+01      9.030₁₀-03
      92     1.848₁₀+01      6.995₁₀-03
      93     1.888₁₀+01      4.720₁₀-03
```

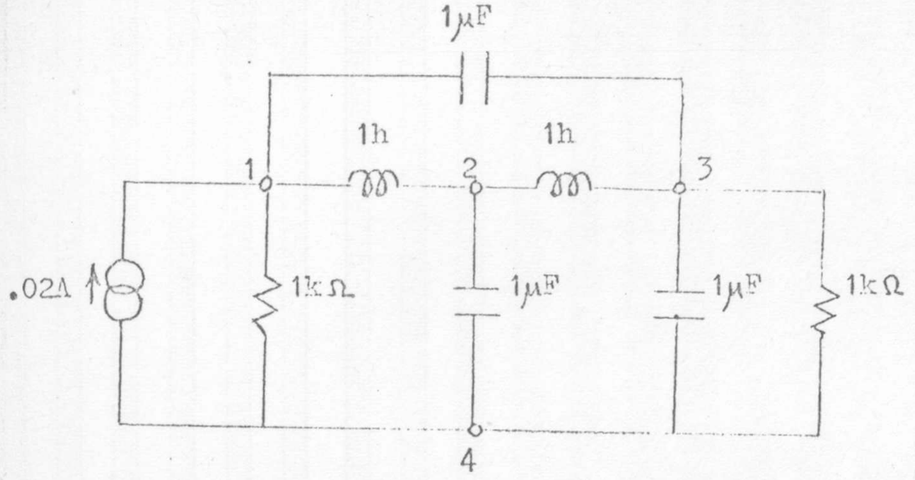
```
       94      1.928_{10}+01       2.582_{10}-03
       95      1.968_{10}+01       8.123_{10}-04
       96      2.008_{10}+01      -4.795_{10}-04
       97      2.048_{10}+01      -1.283_{10}-03
   T=-1.000_{10}+00  NPTS=    98
   NO MORE TRANSIENTS AND STEADY STATE REACHED
```

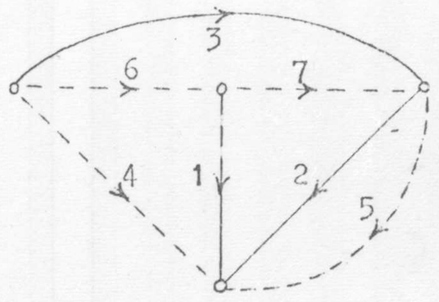


Butterworth Filter

Inductor Current Transient Response (J5)

Ex. 7. Bridged Tee Filter





In this circuit, an independent current source of 0.02 amps is switched on and the transient voltage responses of branches 4, 1 and 2 are observed until they stabilize. The branch voltages settled down after about four milliseconds at ten volts.

This problem was the first to show the need for a variable step size routine, as no suitable single step size could be found over the whole range of interest, without local errors becoming intolerable causing the solution process to fail. The step size ranged from 4×10^{-8} to 4×10^{-4} seconds.

This example was extracted from a paper by Dr. M.W. Bradshaw.[10]

Referring to the plotter outputs in each case the corresponding results extracted from Dr. Bradshaw's paper have been superimposed on the FOFSVM5 outputs and good agreement was obtained between NEETLE outputs and those from FOFSVM5. Such agreement is to be expected, provided the subroutines are operating correctly, as the bridged tee network has a simple current step input.

```
BRIDGED  TEE  FILTER
FOFSVM5 OUTPUT
NRUN=    3

INPUT FOFSVM5.

B=    7   RSOURCE=      1   NY=     3
    B    MN1[B]  MN2[B]  MCOMP[B] ACOMP[B]

    1       1       4      2    1.000₁₀+03
    2       2       4      1    1.000₁₀-06
    3       3       4      1    1.000₁₀-06
    4       1       3      1    1.000₁₀-06
    5       3       4      2    1.000₁₀+03
    6       1       2      3    1.000₁₀+00
    7       2       3      3    1.000₁₀+00
      SMN1    SMN2  SMCOMP   SCOL  MSOURCE
         4       1      0       1     0

N=    3    L=    4    R=    2    C=    3


OUTPUT FOFSVM5;
ORDER=   5    LC=    0    LR=    0    LL=    0    NC=    0    NR=    0    NL=    0

BETAF[L,N]=
     0    -1    -1
     0    -1     0
     1    -1    -1
    -1     1     0

OUTPUT  FOFSVM5;


TR1[N]=         2         3         4

TL1[L]=         1         5         6         7
NETWORK  BRANCHES  REORDERED  INTO  TREE  BRANCHES  AND  LINKS


         B   MN1[B] MN2[B] MCOMP[B]  ACOMP[B]
         1       2       4       1    1.000₁₀-06    V[    1]
         2       3       4       2    1.000₁₀-06    V[    2]
         3       1       3       3    1.000₁₀-06    V[    3]
         4       1       4      51    1.000₁₀+03    J[    4]
         5       3       4      52    1.000₁₀+03    J[    5]
         6       1       2     101    1.000₁₀+00    J[    6]
         7       2       3     102    1.000₁₀+00    J[    7]
```

```
  SMN1  SMN2  SMCOMP  SCOL  MSOURCE
     4     1       0     4        0

SMALLEST OVER LARGEST PIVOT= 1.000⏨+00
THE  STATE  VARIABLES  ARE
(  V[  1]  V[  2]  V[  3]  J[  6]  J[  7])

A=
   0.000⏨+00   0.000⏨+00   0.000⏨+00   1.000⏨+06

   0.000⏨+00  -2.000⏨+03  -1.000⏨+03  -1.000⏨+06

   0.000⏨+00  -1.000⏨+03  -1.000⏨+03  -1.000⏨+06

  -1.000⏨+00   1.000⏨+00   1.000⏨+00   0.000⏨+00

   1.000⏨+00  -1.000⏨+00   0.000⏨+00   0.000⏨+00

B=
   0.000⏨+00

   1.000⏨+06

   1.000⏨+06

   0.000⏨+00

   0.000⏨+00

C=
   0.000⏨+00   1.000⏨+00   1.000⏨+00   0.000⏨+00

   1.000⏨+00   0.000⏨+00   0.000⏨+00   0.000⏨+00

   0.000⏨+00   1.000⏨+00   0.000⏨+00   0.000⏨+00

D=
   0.000⏨+00

   0.000⏨+00

   0.000⏨+00
```

```
-1.000₁₀+06

 1.000₁₀+06

 0.000₁₀+00

 0.000₁₀+00

 0.000₁₀+00









 0.000₁₀+00

 0.000₁₀+00

 0.000₁₀+00
```

```
WHERE  DX/DT=AX+BU    AND  Y=CX+DU

H= 4.000₁₀-08
NINT=       1
TINT= 0.000₁₀+00   UINT= 2.000₁₀-02
TINT= 1.000₁₀+06   UINT= 2.000₁₀-02
EPS= 1.000₁₀-06  EPS1= 2.500₁₀-02  EPS2= 1.000₁₀-01
T0= 0.000₁₀+00  TN= 1.000₁₀+30  NPTS=        500

  F=0  T= 0.000₁₀+00     0.000₁₀+00     0.000₁₀+00     0.000₁₀+00
VMAX= 2.000₁₀+01   IMAX= 2.000₁₀-02

     1    4.000₁₀-08    1.600₁₀-03    6.400₁₀-13    8.000₁₀-04
     2    8.000₁₀-08    3.200₁₀-03    5.120₁₀-12    1.600₁₀-03
     3    1.200₁₀-07    4.799₁₀-03    1.728₁₀-11    2.400₁₀-03
     4    1.600₁₀-07    6.399₁₀-03    4.096₁₀-11    3.199₁₀-03
     5    2.000₁₀-07    7.998₁₀-03    7.999₁₀-11    3.999₁₀-03
     6    2.400₁₀-07    9.597₁₀-03    1.382₁₀-10    4.798₁₀-03
     7    2.800₁₀-07    1.120₁₀-02    2.195₁₀-10    5.598₁₀-03
     8    3.200₁₀-07    1.279₁₀-02    3.276₁₀-10    6.397₁₀-03
     9    3.600₁₀-07    1.439₁₀-02    4.664₁₀-10    7.196₁₀-03
    10    4.000₁₀-07    1.599₁₀-02    6.398₁₀-10    7.995₁₀-03
STEP SIZE INCREASED   H= 8.000₁₀-08

    11    4.800₁₀-07    1.919₁₀-02    1.106₁₀-09    9.593₁₀-03
    12    5.600₁₀-07    2.238₁₀-02    1.756₁₀-09    1.119₁₀-02
    13    6.400₁₀-07    2.558₁₀-02    2.620₁₀-09    1.279₁₀-02
    14    7.200₁₀-07    2.877₁₀-02    3.731₁₀-09    1.438₁₀-02
    15    8.000₁₀-07    3.197₁₀-02    5.117₁₀-09    1.598₁₀-02
    16    8.800₁₀-07    3.516₁₀-02    6.811₁₀-09    1.758₁₀-02
    17    9.600₁₀-07    3.835₁₀-02    8.842₁₀-09    1.917₁₀-02
    18    1.040₁₀-06    4.155₁₀-02    1.124₁₀-08    2.077₁₀-02
    19    1.120₁₀-06    4.474₁₀-02    1.404₁₀-08    2.236₁₀-02
    20    1.200₁₀-06    4.793₁₀-02    1.727₁₀-08    2.396₁₀-02
STEP SIZE INCREASED   H= 2.000₁₀-07

    21    1.400₁₀-06    5.590₁₀-02    2.741₁₀-08    2.794₁₀-02
    22    1.600₁₀-06    6.387₁₀-02    4.092₁₀-08    3.192₁₀-02
    23    1.800₁₀-06    7.184₁₀-02    5.825₁₀-08    3.590₁₀-02
    24    2.000₁₀-06    7.980₁₀-02    7.989₁₀-08    3.988₁₀-02
    25    2.200₁₀-06    8.776₁₀-02    1.063₁₀-07    4.385₁₀-02
    26    2.400₁₀-06    9.571₁₀-02    1.380₁₀-07    4.783₁₀-02
    27    2.600₁₀-06    1.037₁₀-01    1.755₁₀-07    5.180₁₀-02
    28    2.800₁₀-06    1.116₁₀-01    2.191₁₀-07    5.577₁₀-02
    29    3.000₁₀-06    1.196₁₀-01    2.695₁₀-07    5.973₁₀-02
    30    3.200₁₀-06    1.275₁₀-01    3.270₁₀-07    6.369₁₀-02
STEP SIZE INCREASED   H= 4.000₁₀-07
```

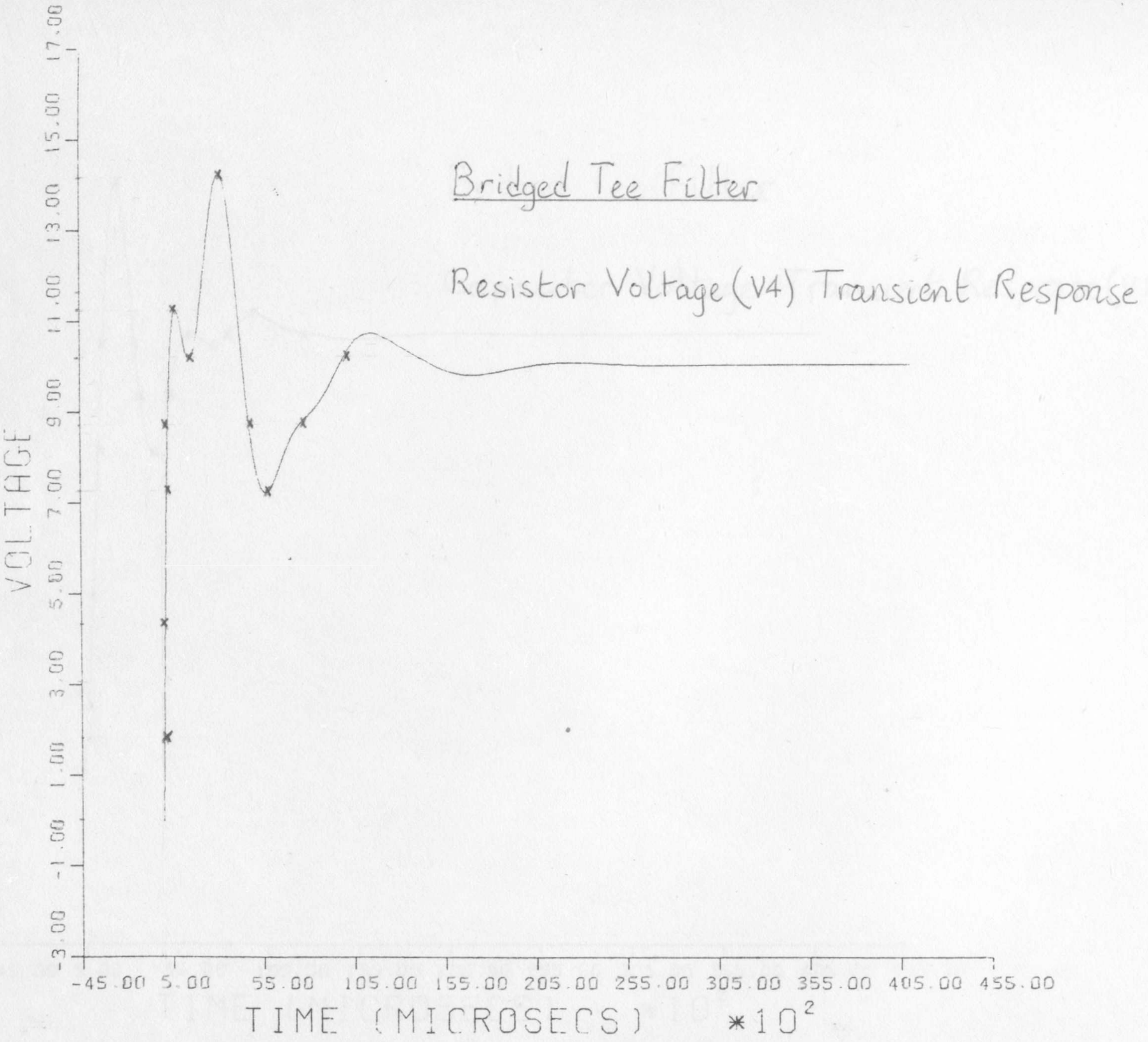
```
    31     3.600₁₀-06     1.434₁₀-01     4.654₁₀-07     7.161₁₀-02
    32     4.000₁₀-06     1.592₁₀-01     6.383₁₀-07     7.952₁₀-02
    33     4.400₁₀-06     1.750₁₀-01     8.493₁₀-07     8.742₁₀-02
    34     4.800₁₀-06     1.909₁₀-01     1.102₁₀-06     9.531₁₀-02
    35     5.200₁₀-06     2.067₁₀-01     1.401₁₀-06     1.032₁₀-01
    36     5.600₁₀-06     2.224₁₀-01     1.750₁₀-06     1.111₁₀-01
    37     6.000₁₀-06     2.382₁₀-01     2.151₁₀-06     1.189₁₀-01
    38     6.400₁₀-06     2.540₁₀-01     2.610₁₀-06     1.268₁₀-01
    39     6.800₁₀-06     2.697₁₀-01     3.130₁₀-06     1.346₁₀-01
    40     7.200₁₀-06     2.854₁₀-01     3.715₁₀-06     1.425₁₀-01
 STEP SIZE INCREASED    H= 8.000₁₀-07

    41     8.000₁₀-06     3.168₁₀-01     5.093₁₀-06     1.581₁₀-01
    42     8.800₁₀-06     3.481₁₀-01     6.775₁₀-06     1.737₁₀-01
    43     9.600₁₀-06     3.794₁₀-01     8.791₁₀-06     1.892₁₀-01
    44     1.040₁₀-05     4.106₁₀-01     1.117₁₀-05     2.048₁₀-01
    45     1.120₁₀-05     4.418₁₀-01     1.394₁₀-05     2.203₁₀-01
    46     1.200₁₀-05     4.728₁₀-01     1.714₁₀-05     2.357₁₀-01
    47     1.280₁₀-05     5.039₁₀-01     2.079₁₀-05     2.511₁₀-01
    48     1.360₁₀-05     5.348₁₀-01     2.493₁₀-05     2.665₁₀-01
    49     1.440₁₀-05     5.657₁₀-01     2.957₁₀-05     2.818₁₀-01
    50     1.520₁₀-05     5.965₁₀-01     3.476₁₀-05     2.971₁₀-01
 STEP SIZE INCREASED    H= 2.000₁₀-06

    51     1.720₁₀-05     6.733₁₀-01     5.030₁₀-05     3.352₁₀-01
    52     1.920₁₀-05     7.498₁₀-01     6.988₁₀-05     3.731₁₀-01
    53     2.120₁₀-05     8.258₁₀-01     9.394₁₀-05     4.107₁₀-01
    54     2.320₁₀-05     9.014₁₀-01     1.229₁₀-04     4.481₁₀-01
    55     2.520₁₀-05     9.767₁₀-01     1.574₁₀-04     4.852₁₀-01
    56     2.720₁₀-05     1.052₁₀+00     1.976₁₀-04     5.221₁₀-01
    57     2.920₁₀-05     1.126₁₀+00     2.441₁₀-04     5.588₁₀-01
    58     3.120₁₀-05     1.200₁₀+00     2.974₁₀-04     5.953₁₀-01
    59     3.320₁₀-05     1.274₁₀+00     3.579₁₀-04     6.315₁₀-01
    60     3.520₁₀-05     1.347₁₀+00     4.260₁₀-04     6.675₁₀-01
 STEP SIZE INCREASED    H= 4.000₁₀-06

    61     3.920₁₀-05     1.493₁₀+00     5.867₁₀-04     7.389₁₀-01
    62     4.320₁₀-05     1.637₁₀+00     7.832₁₀-04     8.093₁₀-01
    63     4.720₁₀-05     1.779₁₀+00     1.019₁₀-03     8.789₁₀-01
    64     5.120₁₀-05     1.920₁₀+00     1.297₁₀-03     9.476₁₀-01
    65     5.520₁₀-05     2.060₁₀+00     1.621₁₀-03     1.015₁₀+00
    66     5.920₁₀-05     2.198₁₀+00     1.994₁₀-03     1.082₁₀+00
    67     6.320₁₀-05     2.335₁₀+00     2.419₁₀-03     1.148₁₀+00
    68     6.720₁₀-05     2.470₁₀+00     2.900₁₀-03     1.213₁₀+00
    69     7.120₁₀-05     2.604₁₀+00     3.440₁₀-03     1.278₁₀+00
    70     7.520₁₀-05     2.736₁₀+00     4.042₁₀-03     1.341₁₀+00
 STEP SIZE INCREASED    H= 8.000₁₀-06
```

```
     71     8.320₁₀-05     2.997₁₀+00     5.445₁₀-03     1.466₁₀+00
     72     9.120₁₀-05     3.252₁₀+00     7.132₁₀-03     1.587₁₀+00
     73     9.920₁₀-05     3.501₁₀+00     9.128₁₀-03     1.705₁₀+00
     74     1.072₁₀-04     3.745₁₀+00     1.146₁₀-02     1.819₁₀+00
     75     1.152₁₀-04     3.984₁₀+00     1.414₁₀-02     1.931₁₀+00
     76     1.232₁₀-04     4.217₁₀+00     1.720₁₀-02     2.039₁₀+00
     77     1.312₁₀-04     4.446₁₀+00     2.065₁₀-02     2.144₁₀+00
     78     1.392₁₀-04     4.669₁₀+00     2.453₁₀-02     2.246₁₀+00
     79     1.472₁₀-04     4.887₁₀+00     2.885₁₀-02     2.345₁₀+00
     80     1.552₁₀-04     5.100₁₀+00     3.362₁₀-02     2.441₁₀+00
 STEP SIZE INCREASED       H= 2.000₁₀-05

     81     1.752₁₀-04     5.611₁₀+00     4.769₁₀-02     2.669₁₀+00
     82     1.952₁₀-04     6.092₁₀+00     6.503₁₀-02     2.879₁₀+00
     83     2.152₁₀-04     6.545₁₀+00     8.591₁₀-02     3.072₁₀+00
     84     2.352₁₀-04     6.970₁₀+00     1.106₁₀-01     3.249₁₀+00
     85     2.552₁₀-04     7.368₁₀+00     1.392₁₀-01     3.410₁₀+00
     86     2.752₁₀-04     7.741₁₀+00     1.721₁₀-01     3.557₁₀+00
     87     2.952₁₀-04     8.089₁₀+00     2.093₁₀-01     3.689₁₀+00
     88     3.152₁₀-04     8.414₁₀+00     2.511₁₀-01     3.807₁₀+00
     89     3.352₁₀-04     8.716₁₀+00     2.975₁₀-01     3.912₁₀+00
     90     3.552₁₀-04     8.996₁₀+00     3.488₁₀-01     4.004₁₀+00
 STEP SIZE INCREASED       H= 4.000₁₀-05

     91     3.952₁₀-04     9.494₁₀+00     4.662₁₀-01     4.154₁₀+00
     92     4.352₁₀-04     9.916₁₀+00     6.039₁₀-01     4.259₁₀+00
     93     4.752₁₀-04     1.027₁₀+01     7.623₁₀-01     4.325₁₀+00
     94     5.152₁₀-04     1.056₁₀+01     9.416₁₀-01     4.355₁₀+00
     95     5.552₁₀-04     1.079₁₀+01     1.142₁₀+00     4.354₁₀+00
     96     5.952₁₀-04     1.097₁₀+01     1.363₁₀+00     4.324₁₀+00
     97     6.352₁₀-04     1.110₁₀+01     1.603₁₀+00     4.270₁₀+00
     98     6.752₁₀-04     1.119₁₀+01     1.864₁₀+00     4.194₁₀+00
     99     7.152₁₀-04     1.125₁₀+01     2.142₁₀+00     4.101₁₀+00
    100     7.552₁₀-04     1.127₁₀+01     2.439₁₀+00     3.992₁₀+00
 STEP SIZE INCREASED       H= 8.000₁₀-05

    101     8.352₁₀-04     1.124₁₀+01     3.081₁₀+00     3.739₁₀+00
    102     9.152₁₀-04     1.114₁₀+01     3.779₁₀+00     3.457₁₀+00
    103     9.952₁₀-04     1.099₁₀+01     4.522₁₀+00     3.164₁₀+00
    104     1.075₁₀-03     1.081₁₀+01     5.298₁₀+00     2.875₁₀+00
    105     1.155₁₀-03     1.063₁₀+01     6.093₁₀+00     2.603₁₀+00
    106     1.235₁₀-03     1.047₁₀+01     6.895₁₀+00     2.358₁₀+00
    107     1.315₁₀-03     1.032₁₀+01     7.692₁₀+00     2.151₁₀+00
    108     1.395₁₀-03     1.022₁₀+01     8.469₁₀+00     1.987₁₀+00
    109     1.475₁₀-03     1.016₁₀+01     9.216₁₀+00     1.870₁₀+00
    110     1.555₁₀-03     1.015₁₀+01     9.923₁₀+00     1.805₁₀+00
    111     1.635₁₀-03     1.018₁₀+01     1.058₁₀+01     1.793₁₀+00
```

```
     112      1.715⏨-03       1.027⏨+01       1.118⏨+01       1.833⏨+00
     113      1.795⏨-03       1.041⏨+01       1.171⏨+01       1.924⏨+00
     114      1.875⏨-03       1.059⏨+01       1.217⏨+01       2.065⏨+00
     115      1.955⏨-03       1.081⏨+01       1.256⏨+01       2.250⏨+00
     116      2.035⏨-03       1.106⏨+01       1.287⏨+01       2.478⏨+00
     117      2.115⏨-03       1.134⏨+01       1.310⏨+01       2.742⏨+00
     118      2.195⏨-03       1.164⏨+01       1.325⏨+01       3.038⏨+00
     119      2.275⏨-03       1.195⏨+01       1.333⏨+01       3.360⏨+00
     120      2.355⏨-03       1.227⏨+01       1.334⏨+01       3.703⏨+00
     121      2.435⏨-03       1.258⏨+01       1.327⏨+01       4.061⏨+00
     122      2.515⏨-03       1.288⏨+01       1.315⏨+01       4.429⏨+00
     123      2.595⏨-03       1.317⏨+01       1.296⏨+01       4.801⏨+00
     124      2.675⏨-03       1.343⏨+01       1.273⏨+01       5.172⏨+00
     125      2.755⏨-03       1.366⏨+01       1.245⏨+01       5.538⏨+00
     126      2.835⏨-03       1.386⏨+01       1.214⏨+01       5.894⏨+00
     127      2.915⏨-03       1.401⏨+01       1.179⏨+01       6.237⏨+00
     128      2.995⏨-03       1.413⏨+01       1.143⏨+01       6.564⏨+00
     129      3.075⏨-03       1.420⏨+01       1.105⏨+01       6.871⏨+00
     130      3.155⏨-03       1.423⏨+01       1.066⏨+01       7.158⏨+00
     131      3.235⏨-03       1.421⏨+01       1.028⏨+01       7.421⏨+00
     132      3.315⏨-03       1.415⏨+01       9.902⏨+00       7.661⏨+00
     133      3.395⏨-03       1.404⏨+01       9.537⏨+00       7.877⏨+00
     134      3.475⏨-03       1.389⏨+01       9.189⏨+00       8.069⏨+00
     135      3.555⏨-03       1.370⏨+01       8.865⏨+00       8.237⏨+00
     136      3.635⏨-03       1.347⏨+01       8.568⏨+00       8.383⏨+00
   STEP SIZE INCREASED        H= 2.000⏨-04

     137      3.835⏨-03       1.277⏨+01       7.961⏨+00       8.657⏨+00
     138      4.035⏨-03       1.192⏨+01       7.572⏨+00       8.825⏨+00
     139      4.235⏨-03       1.101⏨+01       7.406⏨+00       8.923⏨+00
     140      4.435⏨-03       1.010⏨+01       7.442⏨+00       8.985⏨+00
     141      4.635⏨-03       9.259⏨+00       7.646⏨+00       9.044⏨+00
     142      4.835⏨-03       8.534⏨+00       7.970⏨+00       9.126⏨+00
     143      5.035⏨-03       7.958⏨+00       8.363⏨+00       9.250⏨+00
     144      5.235⏨-03       7.546⏨+00       8.776⏨+00       9.423⏨+00
     145      5.435⏨-03       7.298⏨+00       9.166⏨+00       9.645⏨+00
     146      5.635⏨-03       7.198⏨+00       9.502⏨+00       9.907⏨+00
     147      5.835⏨-03       7.223⏨+00       9.763⏨+00       1.020⏨+01
     148      6.035⏨-03       7.344⏨+00       9.940⏨+00       1.049⏨+01
     149      6.235⏨-03       7.527⏨+00       1.004⏨+01       1.078⏨+01
     150      6.435⏨-03       7.745⏨+00       1.006⏨+01       1.105⏨+01
     151      6.635⏨-03       7.970⏨+00       1.004⏨+01       1.127⏨+01
     152      6.835⏨-03       8.183⏨+00       9.977⏨+00       1.144⏨+01
     153      7.035⏨-03       8.371⏨+00       9.903⏨+00       1.156⏨+01
     154      7.235⏨-03       8.530⏨+00       9.834⏨+00       1.162⏨+01
     155      7.435⏨-03       8.658⏨+00       9.784⏨+00       1.162⏨+01
     156      7.635⏨-03       8.761⏨+00       9.763⏨+00       1.158⏨+01
     157      7.835⏨-03       8.847⏨+00       9.773⏨+00       1.150⏨+01
```

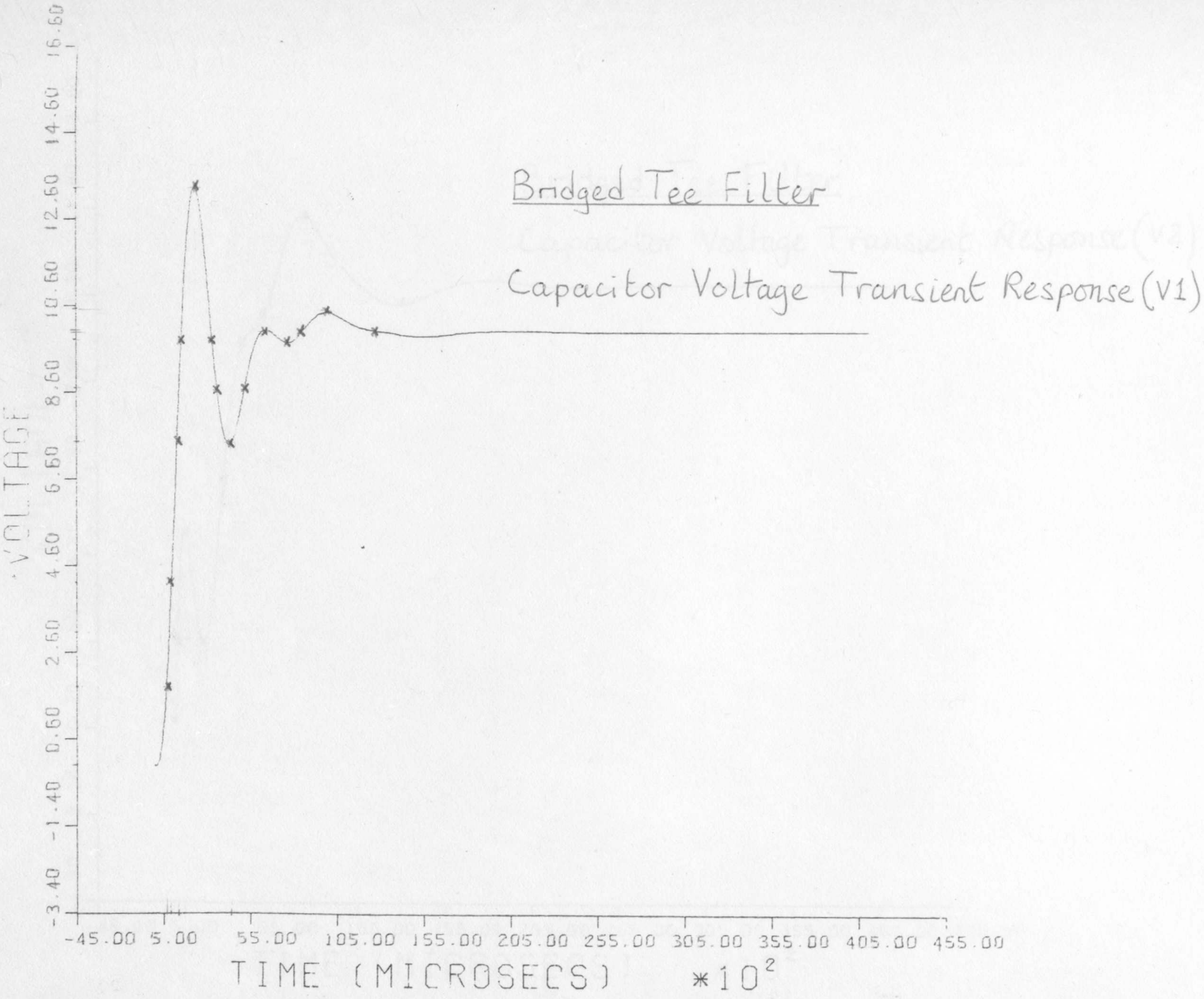
```
     158    8.035₁₀-03      8.925₁₀+00      9.815₁₀+00      1.139₁₀+01
     159    8.235₁₀-03      9.004₁₀+00      9.884₁₀+00      1.126₁₀+01
  STEP SIZE INCREASED      H= 4.000₁₀-04

     160    8.635₁₀-03      9.195₁₀+00      1.007₁₀+01      1.099₁₀+01
     161    9.035₁₀-03      9.450₁₀+00      1.027₁₀+01      1.072₁₀+01
     162    9.435₁₀-03      9.761₁₀+00      1.041₁₀+01      1.050₁₀+01
     163    9.835₁₀-03      1.008₁₀+01      1.046₁₀+01      1.031₁₀+01
     164    1.024₁₀-02      1.037₁₀+01      1.043₁₀+01      1.016₁₀+01
     165    1.064₁₀-02      1.058₁₀+01      1.035₁₀+01      1.001₁₀+01
     166    1.104₁₀-02      1.070₁₀+01      1.025₁₀+01      9.877₁₀+00
     167    1.144₁₀-02      1.073₁₀+01      1.016₁₀+01      9.753₁₀+00
     168    1.184₁₀-02      1.069₁₀+01      1.009₁₀+01      9.652₁₀+00
     169    1.224₁₀-02      1.062₁₀+01      1.005₁₀+01      9.584₁₀+00
     170    1.264₁₀-02      1.053₁₀+01      1.003₁₀+01      9.554₁₀+00
     171    1.304₁₀-02      1.043₁₀+01      1.001₁₀+01      9.562₁₀+00
     172    1.344₁₀-02      1.034₁₀+01      9.982₁₀+00      9.601₁₀+00
     173    1.384₁₀-02      1.024₁₀+01      9.954₁₀+00      9.661₁₀+00
  STEP SIZE INCREASED      H= 8.000₁₀-04

  STEP SIZE DECREASED HMAX= 4.000₁₀-04

     174    1.424₁₀-02      1.015₁₀+01      9.926₁₀+00      9.732₁₀+00
     175    1.464₁₀-02      1.006₁₀+01      9.902₁₀+00      9.805₁₀+00
     176    1.504₁₀-02      9.978₁₀+00      9.889₁₀+00      9.875₁₀+00
     177    1.544₁₀-02      9.905₁₀+00      9.887₁₀+00      9.938₁₀+00
     178    1.584₁₀-02      9.847₁₀+00      9.896₁₀+00      9.992₁₀+00
     179    1.624₁₀-02      9.809₁₀+00      9.913₁₀+00      1.004₁₀+01
     180    1.664₁₀-02      9.789₁₀+00      9.933₁₀+00      1.008₁₀+01
     181    1.704₁₀-02      9.787₁₀+00      9.953₁₀+00      1.011₁₀+01
     182    1.744₁₀-02      9.800₁₀+00      9.971₁₀+00      1.012₁₀+01
     183    1.784₁₀-02      9.822₁₀+00      9.986₁₀+00      1.013₁₀+01
     184    1.824₁₀-02      9.851₁₀+00      9.999₁₀+00      1.013₁₀+01
     185    1.864₁₀-02      9.883₁₀+00      1.001₁₀+01      1.012₁₀+01
     186    1.904₁₀-02      9.916₁₀+00      1.002₁₀+01      1.011₁₀+01
     187    1.944₁₀-02      9.948₁₀+00      1.002₁₀+01      1.009₁₀+01
     188    1.984₁₀-02      9.977₁₀+00      1.003₁₀+01      1.007₁₀+01
     189    2.024₁₀-02      1.000₁₀+01      1.003₁₀+01      1.005₁₀+01
     190    2.064₁₀-02      1.003₁₀+01      1.003₁₀+01      1.003₁₀+01
     191    2.104₁₀-02      1.004₁₀+01      1.003₁₀+01      1.001₁₀+01
     192    2.144₁₀-02      1.006₁₀+01      1.003₁₀+01      9.991₁₀+00
     193    2.184₁₀-02      1.006₁₀+01      1.002₁₀+01      9.978₁₀+00
     194    2.224₁₀-02      1.006₁₀+01      1.002₁₀+01      9.969₁₀+00
     195    2.264₁₀-02      1.006₁₀+01      1.001₁₀+01      9.962₁₀+00
     196    2.304₁₀-02      1.006₁₀+01      1.001₁₀+01      9.959₁₀+00
     197    2.344₁₀-02      1.005₁₀+01      1.000₁₀+01      9.959₁₀+00
     198    2.384₁₀-02      1.004₁₀+01      9.998₁₀+00      9.961₁₀+00
     199    2.424₁₀-02      1.003₁₀+01      9.995₁₀+00      9.965₁₀+00
```

```
   200    2.464⏨-02    1.002⏨+01    9.992⏨+00    9.971⏨+00
   201    2.504⏨-02    1.001⏨+01    9.991⏨+00    9.977⏨+00
   202    2.544⏨-02    1.000⏨+01    9.990⏨+00    9.984⏨+00
   203    2.584⏨-02    9.993⏨+00    9.990⏨+00    9.990⏨+00
   204    2.624⏨-02    9.988⏨+00    9.990⏨+00    9.996⏨+00
   205    2.664⏨-02    9.984⏨+00    9.991⏨+00    1.000⏨+01
   206    2.704⏨-02    9.981⏨+00    9.993⏨+00    1.001⏨+01
   207    2.744⏨-02    9.980⏨+00    9.994⏨+00    1.001⏨+01
   208    2.784⏨-02    9.980⏨+00    9.996⏨+00    1.001⏨+01
   209    2.824⏨-02    9.982⏨+00    9.998⏨+00    1.001⏨+01
   210    2.864⏨-02    9.984⏨+00    9.999⏨+00    1.001⏨+01
   211    2.904⏨-02    9.987⏨+00    1.000⏨+01    1.001⏨+01
   212    2.944⏨-02    9.990⏨+00    1.000⏨+01    1.001⏨+01
   213    2.984⏨-02    9.993⏨+00    1.000⏨+01    1.001⏨+01
   214    3.024⏨-02    9.996⏨+00    1.000⏨+01    1.001⏨+01
   215    3.064⏨-02    9.999⏨+00    1.000⏨+01    1.001⏨+01
   216    3.104⏨-02    1.000⏨+01    1.000⏨+01    1.000⏨+01
   217    3.144⏨-02    1.000⏨+01    1.000⏨+01    1.000⏨+01
   218    3.184⏨-02    1.000⏨+01    1.000⏨+01    1.000⏨+01
   219    3.224⏨-02    1.001⏨+01    1.000⏨+01    9.998⏨+00
   220    3.264⏨-02    1.001⏨+01    1.000⏨+01    9.997⏨+00
   221    3.304⏨-02    1.001⏨+01    1.000⏨+01    9.997⏨+00
   222    3.344⏨-02    1.001⏨+01    1.000⏨+01    9.996⏨+00
   223    3.384⏨-02    1.001⏨+01    1.000⏨+01    9.996⏨+00
   224    3.424⏨-02    1.000⏨+01    1.000⏨+01    9.996⏨+00
   225    3.464⏨-02    1.000⏨+01    1.000⏨+01    9.996⏨+00
   226    3.504⏨-02    1.000⏨+01    9.999⏨+00    9.997⏨+00
   227    3.544⏨-02    1.000⏨+01    9.999⏨+00    9.998⏨+00
   228    3.584⏨-02    1.000⏨+01    9.999⏨+00    9.998⏨+00
   229    3.624⏨-02    1.000⏨+01    9.999⏨+00    9.999⏨+00
   230    3.664⏨-02    9.999⏨+00    9.999⏨+00    9.999⏨+00
   231    3.704⏨-02    9.999⏨+00    9.999⏨+00    1.000⏨+01
   232    3.744⏨-02    9.998⏨+00    9.999⏨+00    1.000⏨+01
   233    3.784⏨-02    9.998⏨+00    9.999⏨+00    1.000⏨+01
   234    3.824⏨-02    9.998⏨+00    1.000⏨+01    1.000⏨+01
   235    3.864⏨-02    9.998⏨+00    1.000⏨+01    1.000⏨+01
   236    3.904⏨-02    9.998⏨+00    1.000⏨+01    1.000⏨+01
   237    3.944⏨-02    9.999⏨+00    1.000⏨+01    1.000⏨+01
   238    3.984⏨-02    9.999⏨+00    1.000⏨+01    1.000⏨+01
   239    4.024⏨-02    9.999⏨+00    1.000⏨+01    1.000⏨+01
   240    4.064⏨-02    1.000⏨+01    1.000⏨+01    1.000⏨+01
   241    4.104⏨-02    1.000⏨+01    1.000⏨+01    1.000⏨+01
 T=-1.000⏨+00  NPTS=  242
 NO MORE TRANSIENTS AND STEADY STATE REACHED
```



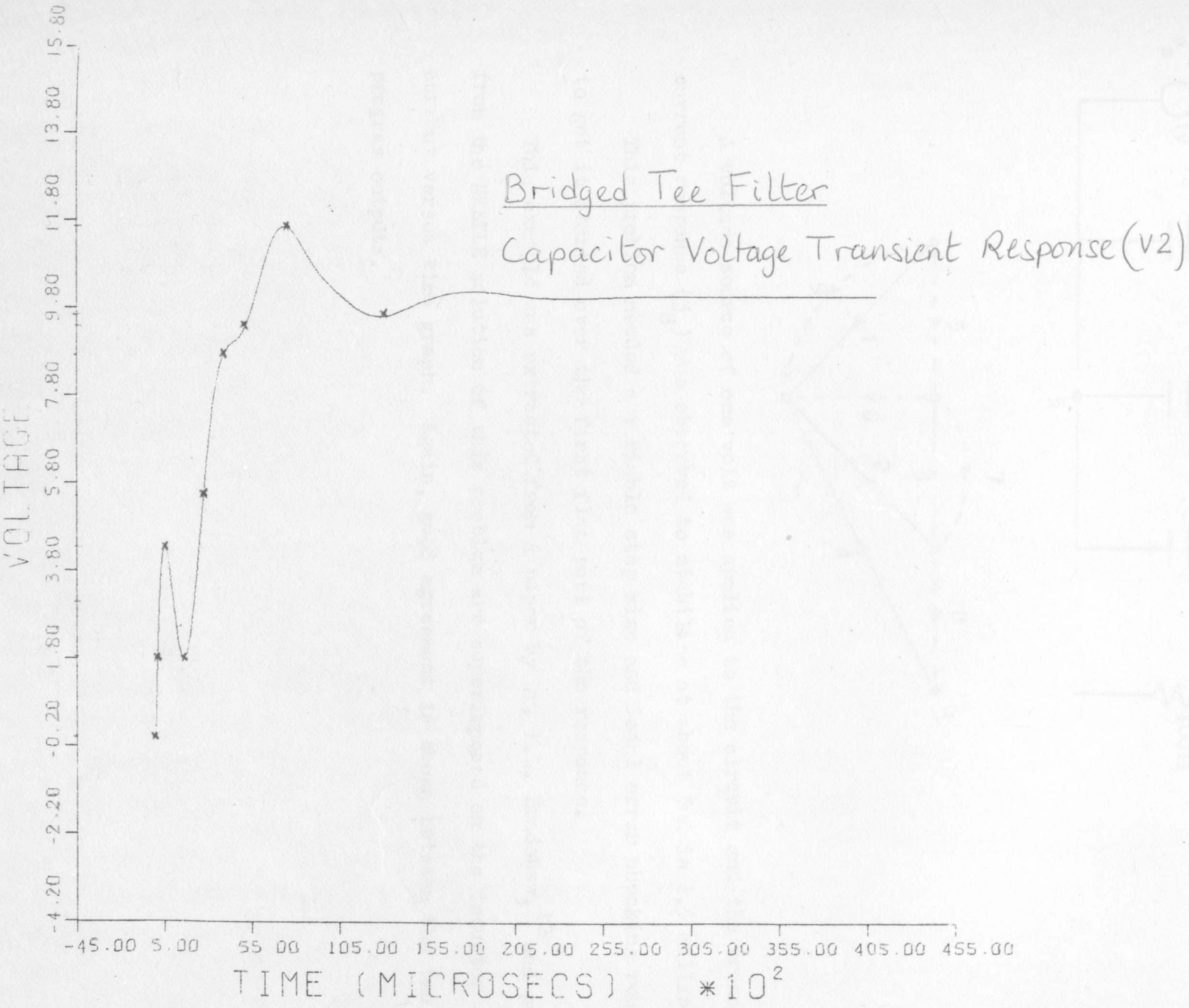
Bridged Tee Filter

Resistor Voltage (V4) Transient Response



Bridged Tee Filter

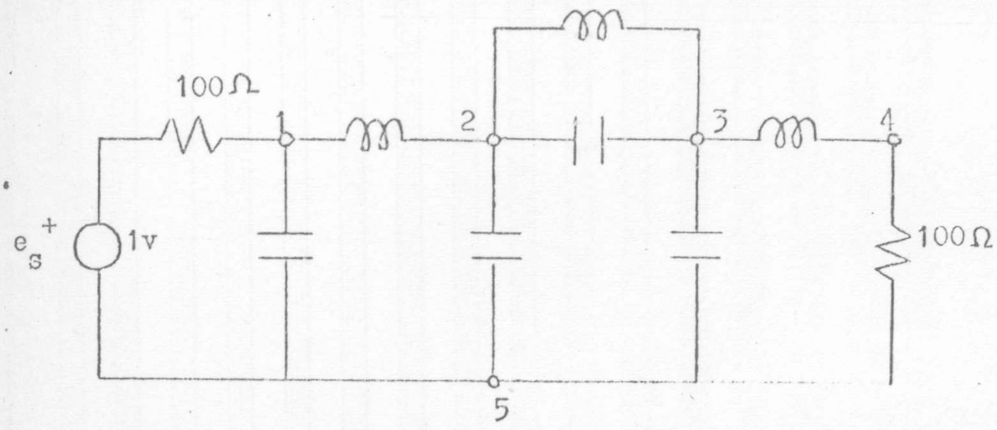
Capacitor Voltage Transient Response (V1)

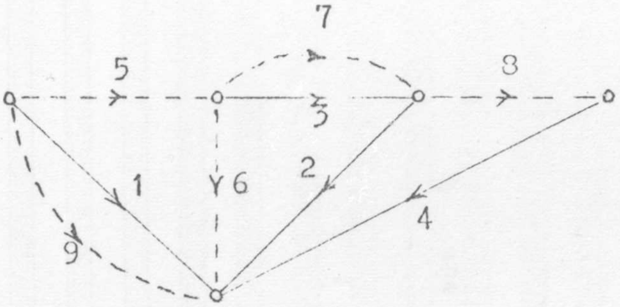


Bridged Tee Filter

Capacitor Voltage Transient Response (V2)

Ex. 8. Step Response of Filter





A voltage source of one volt was applied to the circuit and the transient current response (j_8) was observed to stabilize at about 5mA in 1.5 milliseconds

This problem needed a variable step size and local error checking routines to get it started over the first flat part of the response.

This example was extracted from a paper by Dr. M.W. Bradshaw,[10] and points from the NEETLE solution of this problem are superimposed on the inductor current versus time graph. Again, good agreement is shown between the two program outputs.

```
STEP  RESPONSE  OF  FILTER
FOFSVM5 OUTPUT
NRUN=    4

INPUT FOFSVM5;

B=      9   RSOURCE=     1    NY=      1
   B    MN1[B]  MN2[B]  MCOMP[B] ACOMP[B]

   1       1       5       1     2.799⏨-07
   2       1       2       3     1.654⏨-03
   3       2       5       1     5.434⏨-07
   4       2       3       3     3.973⏨-02
   5       3       5       1     9.360⏨-07
   6       3       4       3     6.881⏨-03
   7       2       3       1     3.010⏨-07
   8       1       5       2     1.000⏨+02
   9       4       5       2     1.000⏨+02
     SMN1    SMN2  SMCOMP    SCOL   MSOURCE
        5       1       1       8       0

N=    4    L=    5    R=    2    C=    4


OUTPUT FOFSVM5;
ORDER=    6    LC=    1    LR=    0    LL=    0    NC=    0    NR=    0    NL=    0

BETAF[L,N]=
    -1     1     1     0
     0    -1    -1     0
     0     0    -1     0
     0    -1     0     1
    -1     0     0     0

OUTPUT  FOFSVM5;



TB1[N]=        1       5       7       9

TL1[L]=        2       3       4       6       8
NETWORK  BRANCHES  REORDERED  INTO  TREE  BRANCHES  AND  LINKS


       B   MN1[B] MN2[B] MCOMP[B] ACOMP[B]
       1       1       5       1    2.799⏨-07    V[    1]
       2       3       5       3    9.360⏨-07    V[    2]
       3       2       3       4    3.010⏨-07    V[    3]
       4       4       5      52    1.000⏨+02    V[    4]
```

```
        5        1        2     101      1.654₁₀-03     J[    5]
        6        2        5       2      5.434₁₀-07     J[    6]
        7        2        3     102      3.973₁₀-02     J[    7]
        8        3        4     103      6.881₁₀-03     J[    8]
        9        1        5      51      1.000₁₀+02     J[    9]

     SMN1  SMN2  SMCOMP  SCOL  MSOURCE
        5       1       1      9       0

  SMALLEST OVER LARGEST PIVOT= 1.000₁₀+00
  SMALLEST OVER LARGEST PIVOT= 1.000₁₀-04
  THE  STATE  VARIABLES  ARE
  (   V[   1]   V[   2]   V[   3]   J[    5]   J[    7]   J[    8])

  A=
     -3.573₁₀+04    0.000₁₀+00    0.000₁₀+00   -3.573₁₀+06    0.000₁₀+00

      0.000₁₀+00    0.000₁₀+00    0.000₁₀+00    3.155₁₀+05    5.696₁₀+05

      0.000₁₀+00    0.000₁₀+00    0.000₁₀+00    9.812₁₀+05   -1.551₁₀+06

      6.046₁₀+02   -6.046₁₀+02   -6.046₁₀+02    0.000₁₀+00    0.000₁₀+00

      0.000₁₀+00    0 000₁₀+00    2.517₁₀+01    0.000₁₀+00    0.000₁₀+00

      0.000₁₀+00    1.453₁₀+02    0.000₁₀+00    0.000₁₀+00    0.000₁₀+00


  B=
      3.573₁₀+04

      0.000₁₀+00

      0.000₁₀+00

      0.000₁₀+00

      0.000₁₀+00

      0.000₁₀+00


  C=
      0.000₁₀+00    0.000₁₀+00    0.000₁₀+00    0.000₁₀+00    0.000₁₀+00


  D=
      0.000₁₀+00
  0.000⏨+00

 -8.852⏨+05

  5.696⏨+05

  0.000⏨+00

  0.000⏨+00

 -1.453⏨+04




        =








  1.000⏨+00
```

```
WHERE  DX/DT=AX+BU   AND  Y=CX+DU

H= 2.000⏨-08
NINT=        1
TINT= 0.000⏨+00   UINT= 1.000⏨+00
TINT= 1.000⏨+06   UINT= 1.000⏨+00
EPS= 1.000⏨-06  EPS1= 2.500⏨-02  EPS2= 1.000⏨-01
T0= 0.000⏨+00  TN= 1.000⏨+30  NPTS=       500

   F=0  T= 0.000⏨+00     0.000⏨+00
VMAX= 1.000⏨+00    IMAX= 1.000⏨-02

       1     2.000⏨-08     6.603⏨-18
       2     4.000⏨-08     1.056⏨-16
       3     6.000⏨-08     5.346⏨-16
       4     8.000⏨-08     1.689⏨-15
       5     1.000⏨-07     4.123⏨-15
       6     1.200⏨-07     8.549⏨-15
       7     1.400⏨-07     1.583⏨-14
       8     1.600⏨-07     2.701⏨-14
       9     1.800⏨-07     4.325⏨-14
      10     2.000⏨-07     6.591⏨-14
STEP SIZE INCREASED    H= 4.000⏨-08

      11     2.400⏨-07     1.366⏨-13
      12     2.800⏨-07     2.530⏨-13
      13     3.200⏨-07     4.314⏨-13
      14     3.600⏨-07     6.908⏨-13
      15     4.000⏨-07     1.052⏨-12
      16     4.400⏨-07     1.540⏨-12
      17     4.800⏨-07     2.180⏨-12
      18     5.200⏨-07     3.002⏨-12
      19     5.600⏨-07     4.036⏨-12
      20     6.000⏨-07     5.317⏨-12
STEP SIZE INCREASED    H= 1.000⏨-07

      21     7.000⏨-07     9.840⏨-12
      22     8.000⏨-07     1.677⏨-11
      23     9.000⏨-07     2.684⏨-11
      24     1.000⏨-06     4.086⏨-11
      25     1.100⏨-06     5.976⏨-11
      26     1.200⏨-06     8.455⏨-11
      27     1.300⏨-06     1.163⏨-10
      28     1.400⏨-06     1.563⏨-10
      29     1.500⏨-06     2.058⏨-10
```

```
     30     1.600⏨-06     2.661⏨-10
 STEP SIZE INCREASED      H= 2.000⏨-07

     31     1.800⏨-06     4.254⏨-10
     32     2.000⏨-06     6.470⏨-10
     33     2.200⏨-06     9.454⏨-10
     34     2.400⏨-06     1.336⏨-09
     35     2.600⏨-06     1.836⏨-09
     36     2.800⏨-06     2.465⏨-09
     37     3.000⏨-06     3.242⏨-09
     38     3.200⏨-06     4.187⏨-09
     39     3.400⏨-06     5.325⏨-09
     40     3.600⏨-06     6.679⏨-09
 STEP SIZE INCREASED      H= 4.000⏨-07

     41     4.000⏨-06     1.014⏨-08
     42     4.400⏨-06     1.478⏨-08
     43     4.800⏨-06     2.084⏨-08
     44     5.200⏨-06     2.858⏨-08
     45     5.600⏨-06     3.827⏨-08
     46     6.000⏨-06     5.021⏨-08
     47     6.400⏨-06     6.471⏨-08
     48     6.800⏨-06     8.210⏨-08
     49     7.200⏨-06     1.027⏨-07
     50     7.600⏨-06     1.270⏨-07
 STEP SIZE INCREASED      H= 1.000⏨-06

     51     8.600⏨-06     2.058⏨-07
     52     9.600⏨-06     3.159⏨-07
     53     1.060⏨-05     4.640⏨-07
     54     1.160⏨-05     6.576⏨-07
     55     1.260⏨-05     9.045⏨-07
     56     1.360⏨-05     1.213⏨-06
     57     1.460⏨-05     1.591⏨-06
     58     1.560⏨-05     2.048⏨-06
     59     1.660⏨-05     2.592⏨-06
     60     1.760⏨-05     3.234⏨-06
 STEP SIZE INCREASED      H= 2.000⏨-06

     61     1.960⏨-05     4.845⏨-06
     62     2.160⏨-05     6.956⏨-06
     63     2.360⏨-05     9.644⏨-06
     64     2.560⏨-05     1.298⏨-05
     65     2.760⏨-05     1.704⏨-05
     66     2.960⏨-05     2.189⏨-05
     67     3.160⏨-05     2.759⏨-05
     68     3.360⏨-05     3.421⏨-05
     69     3.560⏨-05     4.178⏨-05
```

```
        70      3.760⏨-05       5.036⏨-05
        71      3.960⏨-05       5.998⏨-05
        72      4.160⏨-05       7.068⏨-05
        73      4.360⏨-05       8.246⏨-05
        74      4.560⏨-05       9.535⏨-05
        75      4.760⏨-05       1.093⏨-04
        76      4.960⏨-05       1.244⏨-04
        77      5.160⏨-05       1.406⏨-04
        78      5.360⏨-05       1.579⏨-04
        79      5.560⏨-05       1.761⏨-04
        80      5.760⏨-05       1.954⏨-04
        81      5.960⏨-05       2.156⏨-04
        82      6.160⏨-05       2.368⏨-04
        83      6.360⏨-05       2.588⏨-04
   STEP SIZE INCREASED          H= 4.000⏨-06

        84      6.760⏨-05       3.050⏨-04
        85      7.160⏨-05       3.539⏨-04
        86      7.560⏨-05       4.048⏨-04
        87      7.960⏨-05       4.571⏨-04
        88      8.360⏨-05       5.102⏨-04
        89      8.760⏨-05       5.634⏨-04
        90      9.160⏨-05       6.163⏨-04
        91      9.560⏨-05       6.683⏨-04
        92      9.960⏨-05       7.190⏨-04
        93      1.036⏨-04       7.679⏨-04
        94      1.076⏨-04       8.149⏨-04
        95      1.116⏨-04       8.596⏨-04
        96      1.156⏨-04       9.020⏨-04
        97      1.196⏨-04       9.420⏨-04
        98      1.236⏨-04       9.796⏨-04
        99      1.276⏨-04       1.015⏨-03
       100      1.316⏨-04       1.048⏨-03
       101      1.356⏨-04       1.079⏨-03
       102      1.396⏨-04       1.109⏨-03
       103      1.436⏨-04       1.137⏨-03
       104      1.476⏨-04       1.164⏨-03
       105      1.516⏨-04       1.189⏨-03
       106      1.556⏨-04       1.214⏨-03
       107      1.596⏨-04       1.239⏨-03
       108      1.636⏨-04       1.263⏨-03
       109      1.676⏨-04       1.287⏨-03
   STEP SIZE INCREASED          H= 1.000⏨-05

       110      1.776⏨-04       1.347⏨-03
       111      1.876⏨-04       1.407⏨-03
       112      1.976⏨-04       1.469⏨-03
       113      2.076⏨-04       1.531⏨-03
```

```
       114      2.176₁₀-04      1.593₁₀-03
       115      2.276₁₀-04      1.654₁₀-03
       116      2.376₁₀-04      1.714₁₀-03
       117      2.476₁₀-04      1.772₁₀-03
       118      2.576₁₀-04      1.829₁₀-03
       119      2.676₁₀-04      1.887₁₀-03
       120      2.776₁₀-04      1.945₁₀-03
       121      2.876₁₀-04      2.006₁₀-03
       122      2.976₁₀-04      2.070₁₀-03
       123      3.076₁₀-04      2.138₁₀-03
       124      3.176₁₀-04      2.209₁₀-03
    STEP SIZE INCREASED      H= 2.000₁₀-

    STEP SIZE DECREASED HMAX= 1.000₁₀

       125      3.276₁₀-04      2.283₁₀-03
       126      3.376₁₀-04      2.360₁₀-03
       127      3.476₁₀-04      2.440₁₀-03
       128      3.576₁₀-04      2.523₁₀-03
       129      3.676₁₀-04      2.607₁₀-03
       130      3.776₁₀-04      2.693₁₀-03
       131      3.876₁₀-04      2.780₁₀-03
       132      3.976₁₀-04      2.869₁₀-03
       133      4.076₁₀-04      2.959₁₀-03
       134      4.176₁₀-04      3.051₁₀-03
       135      4.276₁₀-04      3.143₁₀-03
       136      4.376₁₀-04      3.236₁₀-03
       137      4.476₁₀-04      3.329₁₀-03
       138      4.576₁₀-04      3.422₁₀-03
       139      4.676₁₀-04      3.514₁₀-03
       140      4.776₁₀-04      3.606₁₀-03
       141      4.876₁₀-04      3.697₁₀-03
       142      4.976₁₀-04      3.786₁₀-03
       143      5.076₁₀-04      3.873₁₀-03
       144      5.176₁₀-04      3.958₁₀-03
       145      5.276₁₀-04      4.042₁₀-03
       146      5.376₁₀-04      4.123₁₀-03
       147      5.476₁₀-04      4.201₁₀-03
       148      5.576₁₀-04      4.277₁₀-03
       149      5.676₁₀-04      4.350₁₀-03
       150      5.776₁₀-04      4.420₁₀-03
       151      5.876₁₀-04      4.487₁₀-03
       152      5.976₁₀-04      4.550₁₀-03
       153      6.076₁₀-04      4.611₁₀-03
       154      6.176₁₀-04      4.668₁₀-03
       155      6.276₁₀-04      4.722₁₀-03
       156      6.376₁₀-04      4.773₁₀-03
       157      6.476₁₀-04      4.820₁₀-03
```

```
-05
-05
```

```
  158    6.576₁₀-04    4.865₁₀-03
  159    6.676₁₀-04    4.906₁₀-03
  160    6.776₁₀-04    4.943₁₀-03
  161    6.876₁₀-04    4.978₁₀-03
  162    6.976₁₀-04    5.010₁₀-03
  163    7.076₁₀-04    5.039₁₀-03
  164    7.176₁₀-04    5.065₁₀-03
  165    7.276₁₀-04    5.088₁₀-03
  166    7.376₁₀-04    5.109₁₀-03
  167    7.476₁₀-04    5.127₁₀-03
  168    7.576₁₀-04    5.142₁₀-03
  169    7.676₁₀-04    5.156₁₀-03
  170    7.776₁₀-04    5.167₁₀-03
  171    7.876₁₀-04    5.177₁₀-03
  172    7.976₁₀-04    5.184₁₀-03
  173    8.076₁₀-04    5.190₁₀-03
  174    8.176₁₀-04    5.194₁₀-03
  175    8.276₁₀-04    5.197₁₀-03
  176    8.376₁₀-04    5.198₁₀-03
  177    8.476₁₀-04    5.198₁₀-03
  178    8.576₁₀-04    5.197₁₀-03
  179    8.676₁₀-04    5.194₁₀-03
  180    8.776₁₀-04    5.191₁₀-03
  181    8.876₁₀-04    5.187₁₀-03
  182    8.976₁₀-04    5.183₁₀-03
  183    9.076₁₀-04    5.177₁₀-03
  184    9.176₁₀-04    5.171₁₀-03
  185    9.276₁₀-04    5.165₁₀-03
  186    9.376₁₀-04    5.158₁₀-03
  187    9.476₁₀-04    5.151₁₀-03
  188    9.576₁₀-04    5.144₁₀-03
  189    9.676₁₀-04    5.137₁₀-03
  190    9.776₁₀-04    5.129₁₀-03
  191    9.876₁₀-04    5.122₁₀-03
  192    9.976₁₀-04    5.115₁₀-03
  193    1.008₁₀-03    5.107₁₀-03
  194    1.018₁₀-03    5.100₁₀-03
  195    1.028₁₀-03    5.092₁₀-03
  196    1.038₁₀-03    5.085₁₀-03
  197    1.048₁₀-03    5.078₁₀-03
  198    1.058₁₀-03    5.072₁₀-03
  199    1.068₁₀-03    5.065₁₀-03
  200    1.078₁₀-03    5.059₁₀-03
  201    1.088₁₀-03    5.053₁₀-03
  202    1.098₁₀-03    5.047₁₀-03
  203    1.108₁₀-03    5.042₁₀-03
  204    1.118₁₀-03    5.037₁₀-03
  205    1.128₁₀-03    5.032₁₀-03
```

```
    206      1.138₁₀-03      5.028₁₀-03
    207      1.148₁₀-03      5.023₁₀-03
    208      1.158₁₀-03      5.019₁₀-03
    209      1.168₁₀-03      5.016₁₀-03
    210      1.178₁₀-03      5.012₁₀-03
    211      1.188₁₀-03      5.009₁₀-03
    212      1.198₁₀-03      5.006₁₀-03
    213      1.208₁₀-03      5.004₁₀-03
    214      1.218₁₀-03      5.001₁₀-03
    215      1.228₁₀-03      4.999₁₀-03
    216      1.238₁₀-03      4.997₁₀-03
    217      1.248₁₀-03      4.995₁₀-03
    218      1.258₁₀-03      4.994₁₀-03
    219      1.268₁₀-03      4.993₁₀-03
    220      1.278₁₀-03      4.992₁₀-03
    221      1.288₁₀-03      4.991₁₀-03
    222      1.298₁₀-03      4.990₁₀-03
    223      1.308₁₀-03      4.989₁₀-03
    224      1.318₁₀-03      4.989₁₀-03
    225      1.328₁₀-03      4.988₁₀-03
    226      1.338₁₀-03      4.988₁₀-03
    227      1.348₁₀-03      4.988₁₀-03
    228      1.358₁₀-03      4.988₁₀-03
    229      1.368₁₀-03      4.988₁₀-03
    230      1.378₁₀-03      4.988₁₀-03
    231      1.388₁₀-03      4.988₁₀-03
    232      1.398₁₀-03      4.988₁₀-03
    233      1.408₁₀-03      4.989₁₀-03
    234      1.418₁₀-03      4.989₁₀-03
    235      1.428₁₀-03      4.989₁₀-03
    236      1.438₁₀-03      4.990₁₀-03
    237      1.448₁₀-03      4.990₁₀-03
    238      1.458₁₀-03      4.991₁₀-03
    239      1.468₁₀-03      4.991₁₀-03
    240      1.478₁₀-03      4.991₁₀-03
    241      1.488₁₀-03      4.992₁₀-03
    242      1.498₁₀-03      4.992₁₀-03
    243      1.508₁₀-03      4.993₁₀-03
    244      1.518₁₀-03      4.993₁₀-03
    245      1.528₁₀-03      4.994₁₀-03
    246      1.538₁₀-03      4.994₁₀-03
    247      1.548₁₀-03      4.995₁₀-03
    248      1.558₁₀-03      4.995₁₀-03
    249      1.568₁₀-03      4.996₁₀-03
    250      1.578₁₀-03      4.996₁₀-03
    251      1.588₁₀-03      4.996₁₀-03
    252      1.598₁₀-03      4.997₁₀-03
    253      1.608₁₀-03      4.997₁₀-03
```

```
    254    1.618⏨-03    4.997⏨-03
    255    1,628⏨-03    4.998⏨-03
    256    1.638⏨-03    4.998⏨-03
    257    1,648⏨-03    4.998⏨-03
    258    1.658⏨-03    4.999⏨-03
    259    1,668⏨-03    4.999⏨-03
    260    1.678⏨-03    4.999⏨-03
 T=-1.000⏨+00  NPTS=  261
  NO MORE TRANSIENTS AND STEADY STATE REACHED
```

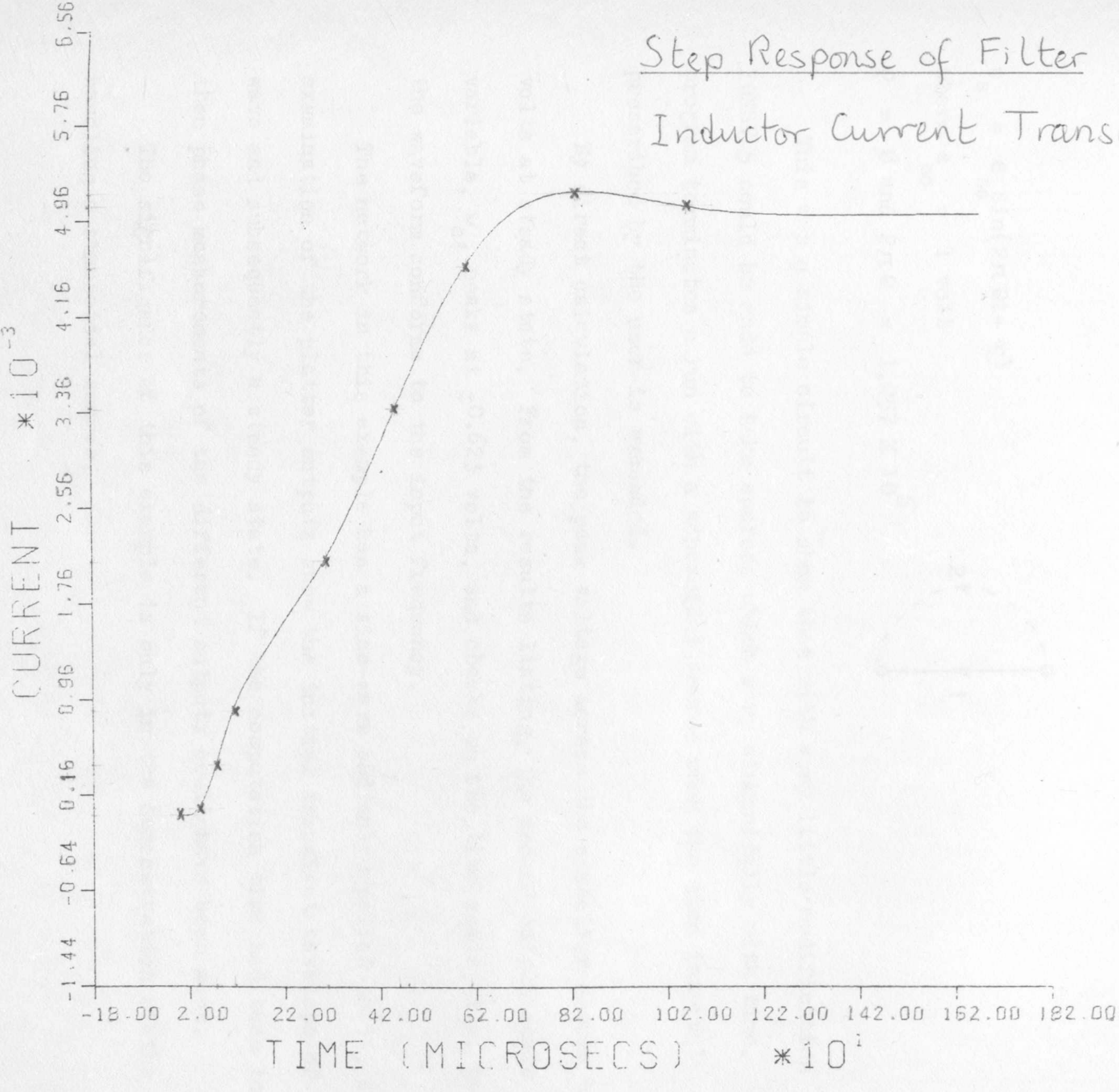
LR

Response of Filter

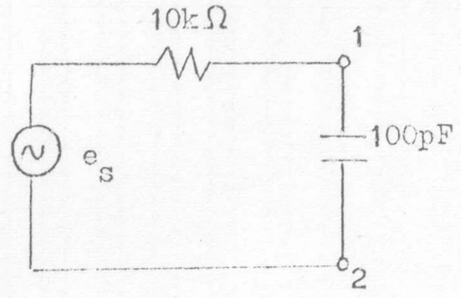
Impulse Response (IR)

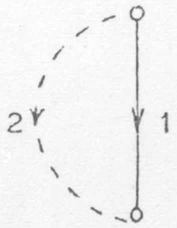
Step Response of Filter

Inductor Current Transient Response (J8)



Ex. 9. Network with Sinusoidal So rce





$e_s = e_{so}\sin(2\pi\theta t+\varphi)$

where $e_{so} = 1$ volt

$\varphi = \emptyset$ and $2\pi\theta = 1.257 \times 10^6$

This was a simple circuit to show that with very little modification FOFSVM5 could be made to take sources which vary sinusoidally with time. The program terminates a run with a sinusoidal source when the time interval prescribed by the user is exceeded.

By direct calculation, the peak voltage across the capacitor should be 0.62 volts at steady state. From the results listing, the second output state variable, v_c, peaks at ±0.623 volts, and checks on the time scale, i.e. period c the waveform conforms to the input frequency..

The network in this example has a sine-wave suddenly applied to it, and examination of the plotter outputs show the initial transient envelope of the si wave and subsequently a steady state. If the computation time had been longer, then phase measurements of the different outputs could have been made.

The significance of this example is only in the demonstration of the progra handling a sinusoidal source.

```
NETWORK  WITH  SINUSOIDAL  SOURCE
FOFSVM5 OUTPUT
NRUN=    6

INPUT FOFSVM5;

B=      2   RSOURCE=      1    NY=      2
   B   MN1[B]  MN2[B]  MCOMP[B] ACOMP[B]

   1      1        2        1     1.000⏨-10
   2      1        2        2     1.000⏨+04
    SMN1    SMN2  SMCOMP    SCOL  MSOURCE
       2        1        1        2       3

N=   1   L=   1   R=   1   C=   1


OUTPUT FOFSVM5;
ORDER=   1   LC=   0    LR=   0    LL=   0   NC=   0    NR=   0    NL=

BETAF[L,N]=
   -1

OUTPUT  FOFSVM5;


TR1[N]=        1

TL1[L]=        2
NETWORK  BRANCHES  REORDERED  INTO  TREE  BRANCHES  AND  LINKS


       B  MN1[B] MN2[B] MCOMP[B] ACOMP[B]
       1     1      2       1     1.000⏨-10   V[    1]
       2     1      2      51     1.000⏨+04   J[    2]

    SMN1  SMN2  SMCOMP  SCOL  MSOURCE
      2      1        1       2        3

SMALLEST OVER LARGEST PIVOT= 1.000⏨+00
THE  STATE  VARIABLES  ARE
(   V[   1])

A=
   -1.000⏨+06


B=
```

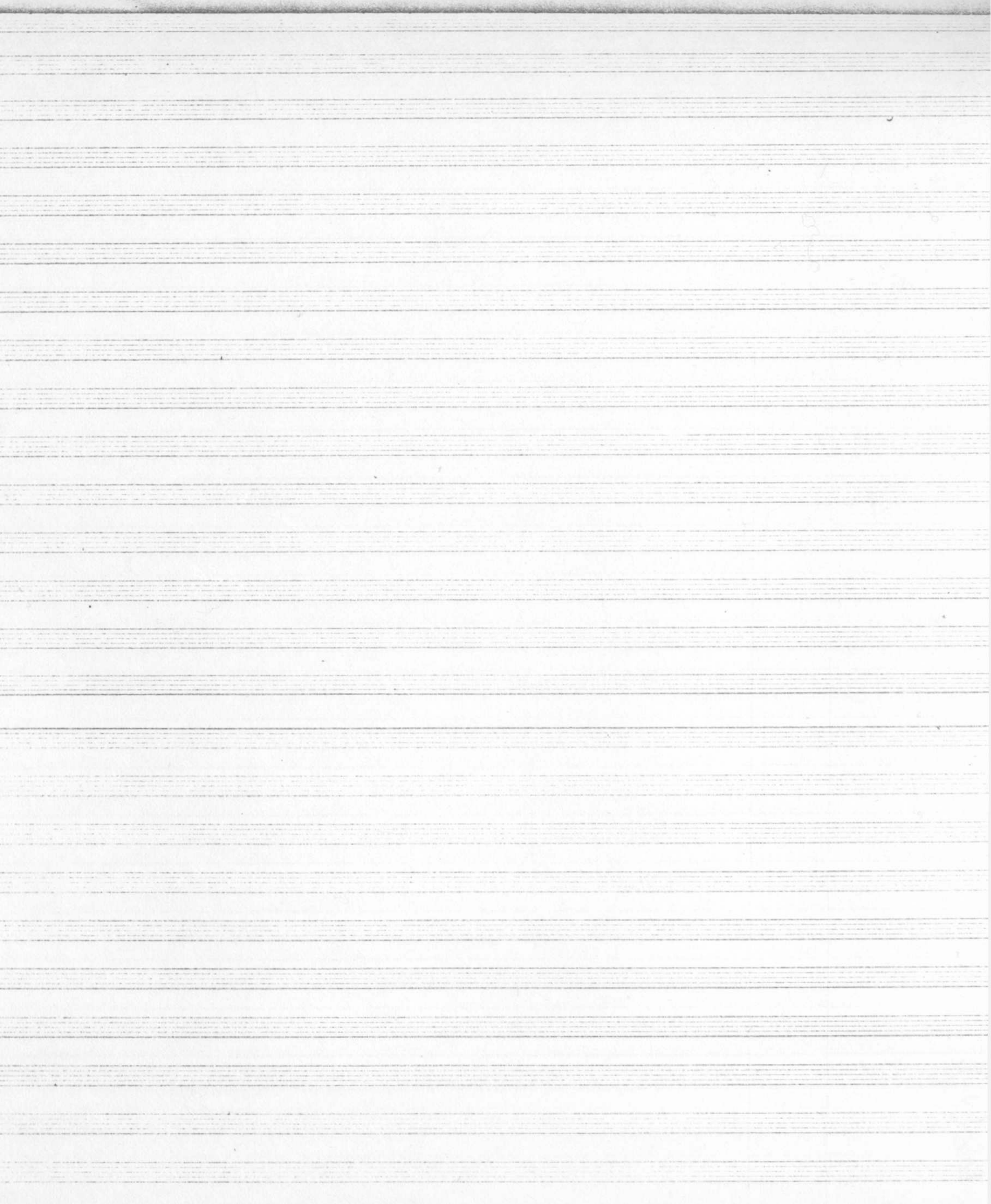
0

```
   1.000⏨+06


C=
  -1.000⏨+00

   1.000⏨+00


D=
   1.000⏨+00

   0.000⏨+00



WHERE  DX/DT=AX+BU   AND  Y=CX+DU

H= 1.000⏨-07
IS= 1.000⏨+00  THETA= 1.257⏨+06  AUX= 0.000⏨+00
EPS= 1.000⏨-06  EPS1= 2.500⏨-02  EPS2= 1.000⏨-01
T0= 0.000⏨+00  TN= 5.000⏨-05  NPTS=     300

  F=0  T= 0.000⏨+00      0.000⏨+00      0.000⏨+00      0.000⏨+00
VMAX= 1.000⏨+00    IMAX= 1.000⏨-04

U[SIN]= 0.000⏨+00
    1    1.000⏨-07    0.000000⏨+00    0.000⏨+00    0.000⏨+00
U[SIN]= 1.253⏨-01
    2    2.000⏨-07    1.192857⏨-02    1.134⏨-01    1.193⏨-02
U[SIN]= 2.487⏨-01
    3    3.000⏨-07    3.446240⏨-02    2.143⏨-01    3.446⏨-02
U[SIN]= 3.682⏨-01
    4    4.000⏨-07    6.621888⏨-02    3.020⏨-01    6.622⏨-02
U[SIN]= 4.818⏨-01
    5    5.000⏨-07    1.057677⏨-01    3.760⏨-01    1.058⏨-01
U[SIN]= 5.879⏨-01
    6    6.000⏨-07    1.516440⏨-01    4.362⏨-01    1.516⏨-01
U[SIN]= 6.846⏨-01
    7    7.000⏨-07    2.023632⏨-01    4.823⏨-01    2.024⏨-01
U[SIN]= 7.706⏨-01
    8    8.000⏨-07    2.564368⏨-01    5.141⏨-01    2.564⏨-01
U[SIN]= 8.444⏨-01
    9    9.000⏨-07    3.123887⏨-01    5.320⏨-01    3.124⏨-01
U[SIN]= 9.049⏨-01
   10    1.000⏨-06    3.687726⏨-01    5.361⏨-01    3.688⏨-01
U[SIN]= 9.511⏨-01
   11    1.100⏨-06    4.241891⏨-01    5.269⏨-01    4.242⏨-01
```

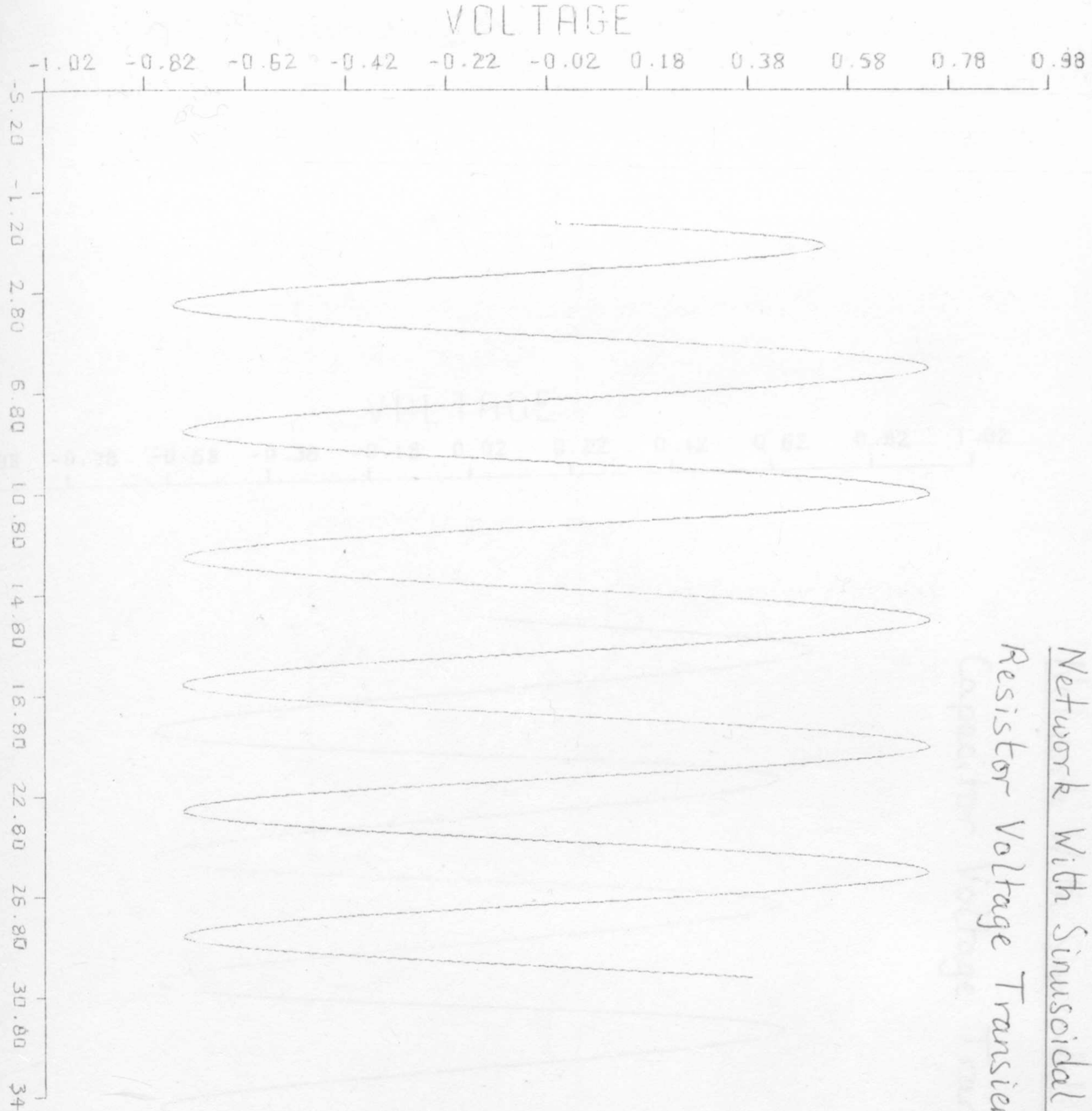
```
U[SIN]= 9.823⏨-01
   12     1.200⏨-06      4.773024⏨-01     5.050⏨-01     4.773⏨-01
U[SIN]= 9.980⏨-01
   13     1.300⏨-06      5.268571⏨-01     4.712⏨-01     5.269⏨-01
U[SIN]= 9.980⏨-01
   14     1.400⏨-06      5.716936⏨-01     4.263⏨-01     5.717⏨-01
U[SIN]= 9.822⏨-01
   15     1.500⏨-06      6.107627⏨-01     3.715⏨-01     6.108⏨-01
U[SIN]= 9.510⏨-01
   16     1,600⏨-06      6.431388⏨-01     3.078⏨-01     6.431⏨-01
U[SIN]= 9.047⏨-01
   17     1.700⏨-06      6.680312⏨-01     2.367⏨-01     6.680⏨-01
U[SIN]= 8.442⏨-01
   18     1.800⏨-06      6.847940⏨-01     1.594⏨-01     6.848⏨-01
U[SIN]= 7.703⏨-01
   19     1.900⏨-06      6.929335⏨-01     7.739⏨-02     6.929⏨-01
U[SIN]= 6.843⏨-01
   20     2.000⏨-06      6.921140⏨-01    -7.793⏨-03     6.921⏨-01
U[SIN]= 5.875⏨-01
   21     2.100⏨-06      6.821608⏨-01    -9.464⏨-02     6.822⏨-01
U[SIN]= 4.815⏨-01
   22     2.200⏨-06      6.630611⏨-01    -1.816⏨-01     6.631⏨-01
U[SIN]= 3.678⏨-01
   23     2.300⏨-06      6.349625⏨-01    -2.672⏨-01     6.350⏨-01
U[SIN]= 2.483⏨-01
   24     2.400⏨-06      5.981693⏨-01    -3.498⏨-01     5.982⏨-01
U[SIN]= 1.249⏨-01
   25     2.500⏨-06      5.531361⏨-01    -4.282⏨-01     5.531⏨-01
U[SIN]=-4.073⏨-04
   26     2.600⏨-06      5.004595⏨-01    -5.009⏨-01     5.005⏨-01
U[SIN]=-1.258⏨-01
   27     2.700⏨-06      4.408675⏨-01    -5.666⏨-01     4.409⏨-01
U[SIN]=-2.491⏨-01
   28     2.800⏨-06      3.752069⏨-01    -6.243⏨-01     3.752⏨-01
U[SIN]=-3.685⏨-01
   29     2.900⏨-06      3.044293⏨-01    -6.730⏨-01     3.044⏨-01
U[SIN]=-4.822⏨-01
   30     3.000⏨-06      2.295747⏨-01    -7.117⏨-01     2.296⏨-01
U[SIN]=-5.882⏨-01
   31     3.100⏨-06      1.517550⏨-01    -7.399⏨-01     1.518⏨-01
U[SIN]=-6.849⏨-01
   32     3.200⏨-06      7.213532⏨-02    -7.571⏨-01     7.214⏨-02
U[SIN]=-7.708⏨-01
   33     3.300⏨-06     -8.084909⏨-03    -7.628⏨-01    -8.085⏨-03
U[SIN]=-8.446⏨-01
   34     3.400⏨-06     -8.769136⏨-02    -7.569⏨-01    -8.769⏨-02
U[SIN]=-9.051⏨-01
   35     3.500⏨-06     -1.654745⏨-01    -7.396⏨-01    -1.655⏨-01
```

```
U[SIN]=-9.512₁₀-01
   36     3.600₁₀-06    -2.402493₁₀-01    -7.110₁₀-01    -2.402₁₀-01
U[SIN]=-9.824₁₀-01
   37     3.700₁₀-06    -3.108740₁₀-01    -6.715₁₀-01    -3.109₁₀-01
U[SIN]=-9.981₁₀-01
   38     3.800₁₀-06    -3.762689₁₀-01    -6.218₁₀-01    -3.763₁₀-01
U[SIN]=-9.980₁₀-01
   39     3.900₁₀-06    -4.354333₁₀-01    -5.626₁₀-01    -4.354₁₀-01
U[SIN]=-9.822₁₀-01
   40     4.000₁₀-06    -4.874620₁₀-01    -4.947₁₀-01    -4.875₁₀-01
U[SIN]=-9.509₁₀-01
   41     4.100₁₀-06    -5.315597₁₀-01    -4.193₁₀-01    -5.316₁₀-01
U[SIN]=-9.045₁₀-01
   42     4.200₁₀-06    -5.670537₁₀-01    -3.375₁₀-01    -5.671₁₀-01
U[SIN]=-8.440₁₀-01
   43     4.300₁₀-06    -5.934050₁₀-01    -2.506₁₀-01    -5.934₁₀-01
U[SIN]=-7.701₁₀-01
   44     4.400₁₀-06    -6.102166₁₀-01    -1.598₁₀-01    -6.102₁₀-01
U[SIN]=-6.840₁₀-01
   45     4.500₁₀-06    -6.172404₁₀-01    -6.678₁₀-02    -6.172₁₀-01
U[SIN]=-5.872₁₀-01
   46     4.600₁₀-06    -6.143810₁₀-01     2.719₁₀-02    -6.144₁₀-01
U[SIN]=-4.811₁₀-01
   47     4.700₁₀-06    -6.016973₁₀-01     1.206₁₀-01    -6.017₁₀-01
U[SIN]=-3.674₁₀-01
   48     4.800₁₀-06    -5.794022₁₀-01     2.120₁₀-01    -5.794₁₀-01
U[SIN]=-2.479₁₀-01
   49     4.900₁₀-06    -5.478587₁₀-01     2.999₁₀-01    -5.479₁₀-01
U[SIN]=-1.245₁₀-01
   50     5.000₁₀-06    -5.075748₁₀-01     3.830₁₀-01    -5.076₁₀-01
U[SIN]= 8.147₁₀-04
   51     5.100₁₀-06    -4.591952₁₀-01     4.600₁₀-01    -4.592₁₀-01
U[SIN]= 1.262₁₀-01
   52     5.200₁₀-06    -4.034915₁₀-01     5.296₁₀-01    -4.035₁₀-01
U[SIN]= 2.495₁₀-01
   53     5.300₁₀-06    -3.413502₁₀-01     5.909₁₀-01    -3.414₁₀-01
U[SIN]= 3.689₁₀-01
   54     5.400₁₀-06    -2.737584₁₀-01     6.427₁₀-01    -2.738₁₀-01
U[SIN]= 4.825₁₀-01
   55     5.500₁₀-06    -2.017886₁₀-01     6.843₁₀-01    -2.018₁₀-01
U[SIN]= 5.885₁₀-01
   56     5.600₁₀-06    -1.265817₁₀-01     7.151₁₀-01    -1.266₁₀-01
U[SIN]= 6.852₁₀-01
   57     5.700₁₀-06    -4.932936₁₀-02     7.345₁₀-01    -4.933₁₀-02
U[SIN]= 7.711₁₀-01
   58     5.800₁₀-06     2.874528₁₀-02     7.424₁₀-01     2.875₁₀-02
U[SIN]= 8.448₁₀-01
   59     5.900₁₀-06     1.064064₁₀-01     7.384₁₀-01     1.064₁₀-01
```

```
U[SIN]= 1.294⏨-01
  252     2.520⏨-05    -4.047709⏨-01     5.342⏨-01    -4.048⏨-01
U[SIN]= 2.527⏨-01
  253     2.530⏨-05    -3.422076⏨-01     5.949⏨-01    -3.422⏨-01
U[SIN]= 3.720⏨-01
  254     2.540⏨-05    -2.742462⏨-01     6.462⏨-01    -2.742⏨-01
U[SIN]= 4.854⏨-01
  255     2.550⏨-05    -2.019586⏨-01     6.873⏨-01    -2.020⏨-01
U[SIN]= 5.911⏨-01
  256     2.560⏨-05    -1.264851⏨-01     7.176⏨-01    -1.265⏨-01
U[SIN]= 6.876⏨-01
  257     2.570⏨-05    -4.901641⏨-02     7.366⏨-01    -4.902⏨-02
U[SIN]= 7.732⏨-01
  258     2.580⏨-05     2.922551⏨-02     7.440⏨-01     2.923⏨-02
U[SIN]= 8.466⏨-01
  259     2.590⏨-05     1.070064⏨-01     7.396⏨-01     1.070⏨-01
U[SIN]= 9.066⏨-01
  260     2.600⏨-05     1.830993⏨-01     7.235⏨-01     1.831⏨-01
U[SIN]= 9.524⏨-01
  261     2.610⏨-05     2.563039⏨-01     6.961⏨-01     2.563⏨-01
U[SIN]= 9.831⏨-01
  262     2.620⏨-05     3.254654⏨-01     6.576⏨-01     3.255⏨-01
U[SIN]= 9.983⏨-01
  263     2.630⏨-05     3.894927⏨-01     6.088⏨-01     3.895⏨-01
U[SIN]= 9.977⏨-01
  264     2.640⏨-05     4.473759⏨-01     5.504⏨-01     4.474⏨-01
U[SIN]= 9.815⏨-01
  265     2.650⏨-05     4.982019⏨-01     4.833⏨-01     4.982⏨-01
U[SIN]= 9.497⏨-01
  266     2.660⏨-05     5.411690⏨-01     4.085⏨-01     5.412⏨-01
U[SIN]= 9.030⏨-01
  267     2.670⏨-05     5.755992⏨-01     3.274⏨-01     5.756⏨-01
U[SIN]= 8.420⏨-01
  268     2.680⏨-05     6.009496⏨-01     2.410⏨-01     6.009⏨-01
U[SIN]= 7.677⏨-01
  269     2.690⏨-05     6.168202⏨-01     1.509⏨-01     6.168⏨-01
U[SIN]= 6.813⏨-01
  270     2.700⏨-05     6.229606⏨-01     5.838⏨-02     6.230⏨-01
U[SIN]= 5.842⏨-01
  271     2.710⏨-05     6.192741⏨-01    -3.505⏨-02     6.193⏨-01
U[SIN]= 4.779⏨-01
  272     2.720⏨-05     6.058186⏨-01    -1.279⏨-01     6.058⏨-01
U[SIN]= 3.640⏨-01
  273     2.730⏨-05     5.828066⏨-01    -2.188⏨-01     5.828⏨-01
U[SIN]= 2.444⏨-01
  274     2.740⏨-05     5.506010⏨-01    -3.062⏨-01     5.506⏨-01
U[SIN]= 1.209⏨-01
  275     2.750⏨-05     5.097099⏨-01    -3.888⏨-01     5.097⏨-01
```



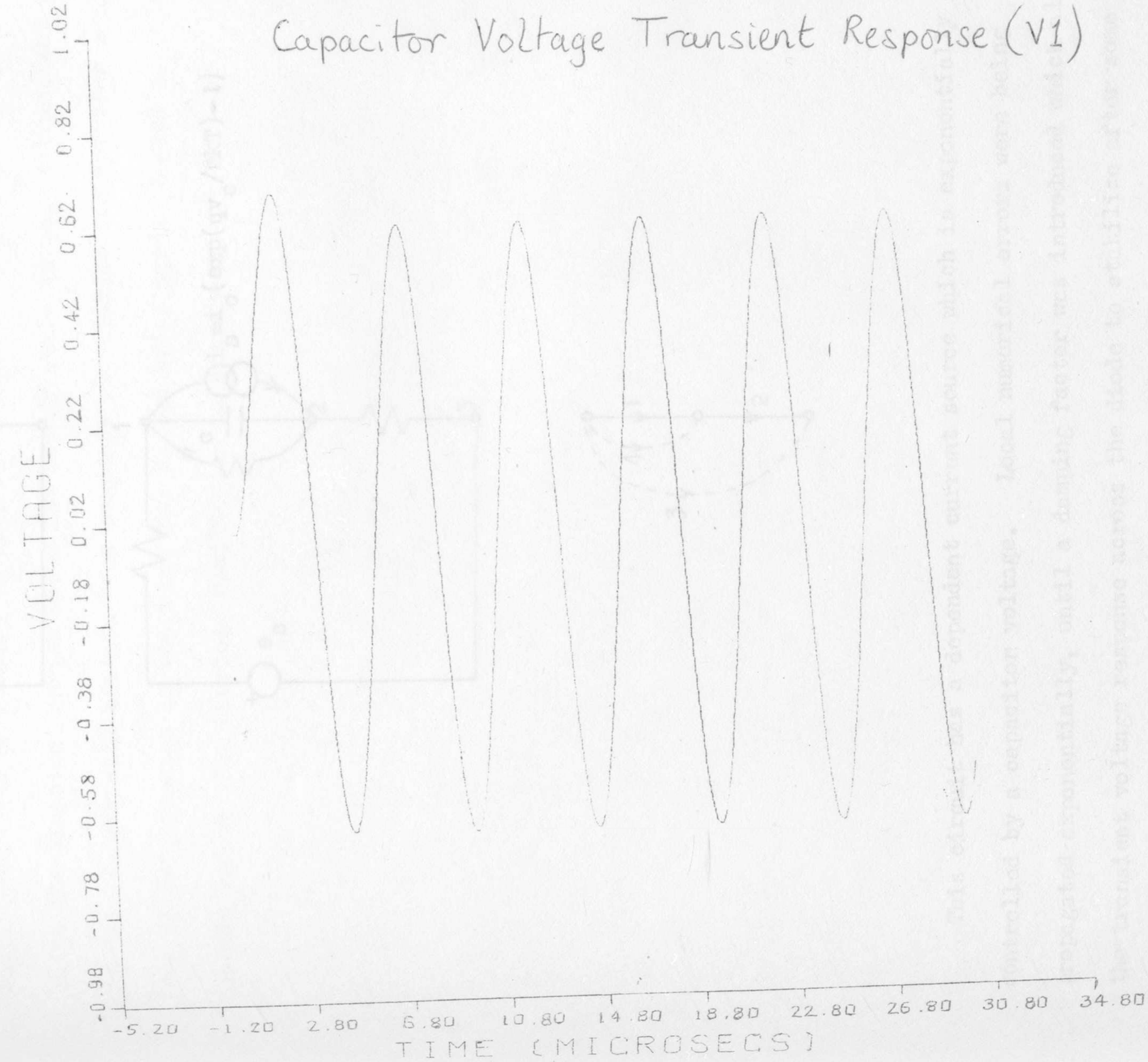
```
U[SIN]=-4.481⏨-03
  276     2.760⏨-05     4.607782⏨-01    -4.653⏨-01     4.608⏨-01
U[SIN]=-1.298⏨-01
  277     2.770⏨-05     4.045779⏨-01    -5.344⏨-01     4.046⏨-01
U[SIN]=-2.531⏨-01
  278     2.780⏨-05     3.419955⏨-01    -5.951⏨-01     3.420⏨-01
U[SIN]=-3.723⏨-01
  279     2.790⏨-05     2.740183⏨-01    -6.464⏨-01     2.740⏨-01
U[SIN]=-4.857⏨-01
  280     2.800⏨-05     2.017184⏨-01    -6.875⏨-01     2.017⏨-01
U[SIN]=-5.915⏨-01
  281     2.810⏨-05     1.262366⏨-01    -7.177⏨-01     1.262⏨-01
U[SIN]=-6.879⏨-01
  282     2.820⏨-05     4.876340⏨-02    -7.366⏨-01     4.876⏨-02
U[SIN]=-7.734⏨-01
  283     2.830⏨-05    -2.947902⏨-02    -7.440⏨-01    -2.948⏨-02
U[SIN]=-8.468⏨-01
  284     2.840⏨-05    -1.072564⏨-01    -7.395⏨-01    -1.073⏨-01
U[SIN]=-9.068⏨-01
  285     2.850⏨-05    -1.833419⏨-01    -7.234⏨-01    -1.833⏨-01
U[SIN]=-9.525⏨-01
  286     2.860⏨-05    -2.565352⏨-01    -6.959⏨-01    -2.565⏨-01
U[SIN]=-9.831⏨-01
  287     2.870⏨-05    -3.256817⏨-01    -6.575⏨-01    -3.257⏨-01
U[SIN]=-9.983⏨-01
  288     2.880⏨-05    -3.896908⏨-01    -6.086⏨-01    -3.897⏨-01
U[SIN]=-9.977⏨-01
  289     2.890⏨-05    -4.475525⏨-01    -5.502⏨-01    -4.476⏨-01
U[SIN]=-9.814⏨-01
  290     2.900⏨-05    -4.983543⏨-01    -4.830⏨-01    -4.984⏨-01
U[SIN]=-9.496⏨-01
  291     2.910⏨-05    -5.412947⏨-01    -4.083⏨-01    -5.413⏨-01
U[SIN]=-9.028⏨-01
  292     2.920⏨-05    -5.756963⏨-01    -3.271⏨-01    -5.757⏨-01
U[SIN]=-8.418⏨-01
  293     2.930⏨-05    -6.010165⏨-01    -2.408⏨-01    -6.010⏨-01
U[SIN]=-7.675⏨-01
  294     2.940⏨-05    -6.168559⏨-01    -1.506⏨-01    -6.169⏨-01
U[SIN]=-6.810⏨-01
  295     2.950⏨-05    -6.229646⏨-01    -5.808⏨-02    -6.230⏨-01
U[SIN]=-5.839⏨-01
  296     2.960⏨-05    -6.192462⏨-01     3.536⏨-02    -6.192⏨-01
U[SIN]=-4.775⏨-01
  297     2.970⏨-05    -6.057593⏨-01     1.282⏨-01    -6.058⏨-01
U[SIN]=-3.636⏨-01
  298     2.980⏨-05    -5.827169⏨-01     2.191⏨-01    -5.827⏨-01
U[SIN]=-2.440⏨-01
  299     2.990⏨-05    -5.504822⏨-01     3.065⏨-01    -5.505⏨-01
```



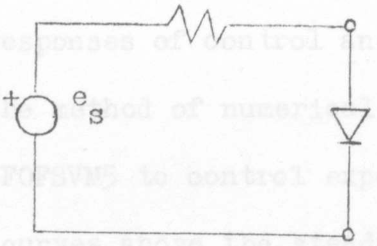
Network With Sinusoidal Source
Resistor Voltage Transient Response (V2)

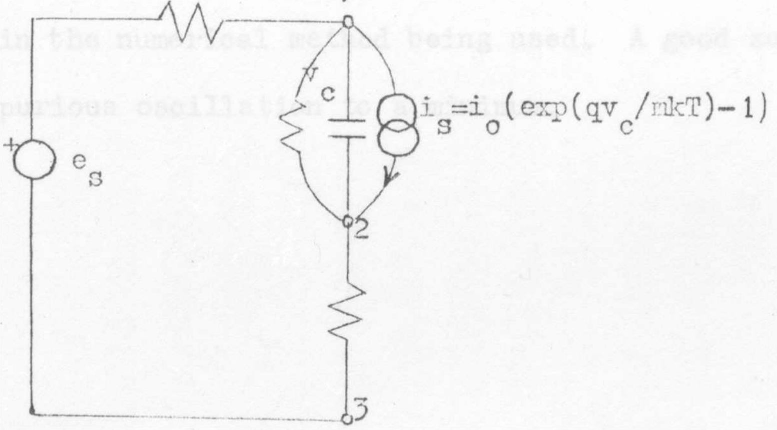
Network With Sinusoidal Source

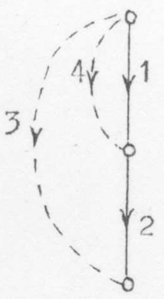
Capacitor Voltage Transient Response (V1)



Ex. 10. Non-linear Model of Diode







This circuit has a dependent current source which is exponentially controlled by a capacitor voltage. Local numerical errors were being propagated exponentially, until a damping factor was introduced which allowed the transient voltage response across the diode to stabilize after some oscillation.

Note that the input voltage e_s is a square pulse, switched from 0 to -4 volts at t = 0, then to +4 volts at t = 10^{-7} seconds.

The transient responses of control and output voltages contain some oscillation due to the method of numerical integration being used. The damping factor, employed by FOFSVM5 to control exponentially propagated locar errors, clips the transient curves above the steady state value. A transient simulated in this manner will contain true response oscillation and spurious oscillation due to deficiencies in the numerical method being used. A good solution method should reduce this spurious oscillation to a minimum.

```
NON-LINEAR  MODEL  OF  DIODE
  FOFSVM5 OUTPUT
  NRUN=   2

 INPUT FOFSVM5;

 B=    4   RSOURCE=    2   NY=    2
    B    MN1[B]  MN2[B]  MCOMP[B] ACOMP[B]

    1       2       1       1     2.000⏨-12
    2       3       2       2     5.000⏨+00
    3       1       3       2     1.000⏨+04
    4       2       1       2     1.000⏨+05
     SMN1    SMN2  SMCOMP   SCOL  MSOURCE
        1       2       0      4       2
        3       1       1      3       0

 N=   2   L=   2   R=   3   C=   1


 OUTPUT FOFSVM5;
 ORDER=   1   LC=   0   LR=   1   LL=   0   NC=   0   NR=   1   NL=   0

 BETAF[L,N]=
    -1   -1
    -1    0

 OUTPUT  FOFSVM5;


 TR1[N]=        1       2

 TL1[L]=        3       4
 NETWORK  BRANCHES  REORDERED  INTO  TREE  BRANCHES  AND  LINKS


        B  MN1[B] MN2[B] MCOMP[B] ACOMP[B]
        1      1      2      1    2.000⏨-12    V[    1]
        2      2      3     51    5.000⏨+00    V[    2]
        3      1      3     52    1.000⏨+04    J[    3]
        4      1      2     53    1.000⏨+05    J[    4]

      SMN1  SMN2  SMCOMP  SCOL  MSOURCE
        1      2      0      4.      2
        3      1      1      3       0

 TB3=
        2      4
 TL3=
        1       3

 BETAF[L,N]=
      0    -1
     -1    -1
 TB4=
        1       2
 TL4=
        3       4

 BETAF[L,N]=
     -1    -1
     -1     0
 SMALLEST OVER LARGEST PIVOT= 1.000₁₀+00
 SMALLEST OVER LARGEST PIVOT= 1.000₁₀+00
 SMALLEST OVER LARGEST PIVOT= 1.000₁₀+00
 THE  STATE  VARIABLES  ARE
 (    V[    1])


 A=
     -5.498₁₀+07


 B=
     -5.000₁₀+11    4.998₁₀+07


 C=
      1.000₁₀+00

      9.995₁₀-01


 D=
      0.000₁₀+00    0.000₁₀+00

      0.000₁₀+00    4,998₁₀-04



 WHERE  DX/DT=AX+BU   AND  Y=CX+DU

 H= 1.000₁₀-09
 NINT=        2
 IS= 1.000₁₀-11  THETA= 2.641₁₀+01
 TINT= 0.000₁₀+00    UINT=-4.000₁₀+00
 TINT= 1.000₁₀-07    UINT=-4.000₁₀+00
```

```
TINT= 1.000₁₀-07   UINT= 4.000₁₀+00
TINT= 1.000₁₀+06   UINT= 4.000₁₀+00
VMAX= 4.000₁₀+00   IMAX= 4.000₁₀-04

EPS= 1.000₁₀-06  EPS1= 2.500₁₀-02  EPS2= 1.000₁₀-01
T0= 0.000₁₀+00  TN= 9.095₁₀-07  NPTS=        300

  F=0  T= 0.000₁₀+00     0.000₁₀+00      0.000₁₀+00      6.076₁₀-67
U[DPT]= 0.000₁₀+00
     1     1.000₁₀-09   -1.945046₁₀-01   -1.945₁₀-01    -1.964₁₀-01
U[DPT]=-8.596₁₀-12
     2     2.000₁₀-09   -3.786048₁₀-01   -3.786₁₀-01    -3.804₁₀-01
U[DPT]=-9.630₁₀-12
     3     3.000₁₀-09   -5.528573₁₀-01   -5.529₁₀-01    -5.546₁₀-01
U[DPT]=-9.864₁₀-12
     4     4.000₁₀-09   -7.177887₁₀-01   -7.178₁₀-01    -7.194₁₀-01
U[DPT]=-9.939₁₀-12
     5     5.000₁₀-09   -8.738976₁₀-01   -8.739₁₀-01    -8.755₁₀-01
U[DPT]=-9.969₁₀-12
     6     6.000₁₀-09   -1.021656₁₀+00   -1.022₁₀+00    -1.023₁₀+00
U[DPT]=-9.982₁₀-12
     7     7.000₁₀-09   -1.161511₁₀+00   -1.162₁₀+00    -1.163₁₀+00
U[DPT]=-9.989₁₀-12
     8     8.000₁₀-09   -1.293884₁₀+00   -1.294₁₀+00    -1.295₁₀+00
U[DPT]=-9.993₁₀-12
     9     9.000₁₀-09   -1.419177₁₀+00   -1.419₁₀+00    -1.420₁₀+00
U[DPT]=-9.995₁₀-12
    10     1.000₁₀-08   -1.537767₁₀+00   -1.538₁₀+00    -1.539₁₀+00
U[DPT]=-9.997₁₀-12
    11     1.100₁₀-08   -1.650014₁₀+00   -1.650₁₀+00    -1.651₁₀+00
U[DPT]=-9.998₁₀-12
    12     1.200₁₀-08   -1.756257₁₀+00   -1.756₁₀+00    -1.757₁₀+00
U[DPT]=-9.998₁₀-12
    13     1.300₁₀-08   -1.856817₁₀+00   -1.857₁₀+00    -1.858₁₀+00
U[DPT]=-9.999₁₀-12
    14     1.400₁₀-08   -1.951997₁₀+00   -1.952₁₀+00    -1.953₁₀+00
U[DPT]=-9.999₁₀-12
    15     1.500₁₀-08   -2.042087₁₀+00   -2.042₁₀+00    -2.043₁₀+00
U[DPT]=-9.999₁₀-12
    16     1.600₁₀-08   -2.127357₁₀+00   -2.127₁₀+00    -2.128₁₀+00
U[DPT]=-9.999₁₀-12
    17     1.700₁₀-08   -2.208066₁₀+00   -2.208₁₀+00    -2.209₁₀+00
U[DPT]=-1.000₁₀-11
    18     1.800₁₀-08   -2.284457₁₀+00   -2.284₁₀+00    -2.285₁₀+00
U[DPT]=-1.000₁₀-11
    19     1.900₁₀-08   -2.356763₁₀+00   -2.357₁₀+00    -2.358₁₀+00
U[DPT]=-1.000₁₀-11
    20     2.000₁₀-08   -2.425200₁₀+00   -2.425₁₀+00    -2.426₁₀+00
```

```
U[DPT]=-1.000⏨-11
   21     2.100⏨-08    -2.489977⏨+00    -2.490⏨+00    -2.491⏨+00
U[DPT]=-1.000⏨-11
   22     2.200⏨-08    -2.551289⏨+00    -2.551⏨+00    -2.552⏨+00
U[DPT]=-1.000⏨-11
   23     2.300⏨-08    -2.609321⏨+00    -2.609⏨+00    -2.610⏨+00
STEP SIZE INCREASED    H= 2.000⏨-09

U[DPT]=-1.000⏨-11
   24     2.500⏨-08    -2.716241⏨+00    -2.716⏨+00    -2.717⏨+00
U[DPT]=-1.000⏨-11
   25     2.700⏨-08    -2.812028⏨+00    -2.812⏨+00    -2.813⏨+00
U[DPT]=-1.000⏨-11
   26     2.900⏨-08    -2.897842⏨+00    -2.898⏨+00    -2.898⏨+00
U[DPT]=-1.000⏨-11
   27     3.100⏨-08    -2.974721⏨+00    -2.975⏨+00    -2.975⏨+00
U[DPT]=-1.000⏨-11
   28     3.300⏨-08    -3.043595⏨+00    -3.044⏨+00    -3.044⏨+00
U[DPT]=-1.000⏨-11
   29     3.500⏨-08    -3.105298⏨+00    -3.105⏨+00    -3.106⏨+00
U[DPT]=-1.000⏨-11
   30     3.700⏨-08    -3.160576⏨+00    -3.161⏨+00    -3.161⏨+00
U[DPT]=-1.000⏨-11
   31     3.900⏨-08    -3.210099⏨+00    -3.210⏨+00    -3.210⏨+00
U[DPT]=-1.000⏨-11
   32     4.100⏨-08    -3.254465⏨+00    -3.254⏨+00    -3.255⏨+00
U[DPT]=-1.000⏨-11
   33     4.300⏨-08    -3.294212⏨+00    -3.294⏨+00    -3.295⏨+00
U[DPT]=-1.000⏨-11
   34     4.500⏨-08    -3.329821⏨+00    -3.330⏨+00    -3.330⏨+00
STEP SIZE INCREASED    H= 5.000⏨-09

U[DPT]=-1.000⏨-11
   35     5.000⏨-08    -3.403451⏨+00    -3.403⏨+00    -3.404⏨+00
U[DPT]=-1.000⏨-11
   36     5.500⏨-08    -3.459386⏨+00    -3.459⏨+00    -3.460⏨+00
U[DPT]=-1.000⏨-11
   37     6.000⏨-08    -3.501878⏨+00    -3.502⏨+00    -3.502⏨+00
U[DPT]=-1.000⏨-11
   38     6.500⏨-08    -3.534157⏨+00    -3.534⏨+00    -3.534⏨+00
U[DPT]=-1.000⏨-11
   39     7.000⏨-08    -3.558679⏨+00    -3.559⏨+00    -3.559⏨+00
U[DPT]=-1.000⏨-11
   40     7.500⏨-08    -3.577307⏨+00    -3.577⏨+00    -3.578⏨+00
U[DPT]=-1.000⏨-11
   41     8.000⏨-08    -3.591459⏨+00    -3.591⏨+00    -3.592⏨+00
U[DPT]=-1.000⏨-11
   42     8.500⏨-08    -3.602209⏨+00    -3.602⏨+00    -3.602⏨+00
```

```
UCDPT]=-1.000₁₀-11
   43     9.000₁₀-08   -3.610375₁₀+00    -3.610₁₀+00    -
UCDPT]=-1.000₁₀-11
   44     9.500₁₀-08   -3.616579₁₀+00    -3.617₁₀+00    -
STEP SIZE INCREASED    H= 1.000₁₀-08

UCDPT]=-1.000₁₀-11
STEP SIZE DECREASED    H= 5.000₁₀-09

UCDPT]=-1.000₁₀-11
STEP SIZE DECREASED    H= 2.000₁₀-09

UCDPT]=-1.000₁₀-11
STEP SIZE DECREASED    H= 1.000₁₀-09

UCDPT]=-1.000₁₀-11
   45     1.130₁₀-07   -3.228618₁₀+00    -3.229₁₀+00    -
UCDPT]=-1.000₁₀-11
   46     1.140₁₀-07   -2.861410₁₀+00    -2.861₁₀+00    -
UCDPT]=-1.000₁₀-11
   47     1.150₁₀-07   -2.513844₁₀+00    -2.514₁₀+00    -
UCDPT]=-1.000₁₀-11
   48     1.160₁₀-07   -2.184870₁₀+00    -2.185₁₀+00    -
UCDPT]=-1.000₁₀-11
   49     1.170₁₀-07   -1.873493₁₀+00    -1.873₁₀+00    -
UCDPT]=-1.000₁₀-11
   50     1.180₁₀-07   -1.578772₁₀+00    -1.579₁₀+00    -
UCDPT]=-1.000₁₀-11
   51     1.190₁₀-07   -1.299817₁₀+00    -1.300₁₀+00    -
UCDPT]=-1.000₁₀-11
   52     1.200₁₀-07   -1.035783₁₀+00    -1.036₁₀+00    -
UCDPT]=-1.000₁₀-11
   53     1.210₁₀-07   -7.858725₁₀-01    -7.859₁₀-01    -
UCDPT]=-1.000₁₀-11
   54     1.220₁₀-07   -5.493304₁₀-01    -5.493₁₀-01    -
UCDPT]=-1.000₁₀-11
   55     1.230₁₀-07   -3.254412₁₀-01    -3.254₁₀-01    -
UCDPT]=-1.000₁₀-11
   56     1.240₁₀-07   -1.135282₁₀-01    -1.135₁₀-01    -
UCDPT]=-9.984₁₀-12
   57     1.250₁₀-07    8.704917₁₀-02     8.705₁₀-02
UCDPT]=-6.872₁₀-12
   58     1.260₁₀-07    2.768974₁₀-01     2.769₁₀-01
UCDPT]= 4.574₁₀-10
   59     1.270₁₀-07    4.565901₁₀-01     4.566₁₀-01
UCDPT]= 5.344₁₀-08
   60     1.280₁₀-07    6.266450₁₀-01     6.266₁₀-01
UCDPT]= 4.741₁₀-06
```

```
-3.611⏨+00

-3.617⏨+00







-3.225⏨+00

-2.858⏨+00

-2.511⏨+00

-2.182⏨+00

-1.871⏨+00

-1.576⏨+00

-1.297⏨+00

-1.033⏨+00

-7.835⏨-01

-5.471⏨-01

-3.233⏨-01

-1.115⏨-01

 8.900⏨-02

 2.788⏨-01

 4.584⏨-01

 6.283⏨-01
```

```
   61    1.290⏨-07    7.853228⏨-01    7.853⏨-01    7.869⏨-01
U[DPT]= 3.119⏨-04
   62    1.300⏨-07    7.860967⏨-01    7.861⏨-01    7.877⏨-01
U[DPT]= 6.137⏨-04
   63    1.310⏨-07    6.399668⏨-01    6.400⏨-01    6.416⏨-01
U[DPT]= 6.019⏨-04
   64    1.320⏨-07    5.073953⏨-01    5.074⏨-01    5.091⏨-01
U[DPT]= 5.843⏨-04
   65    1,330⏨-07    3.904966⏨-01    3.905⏨-01    3.923⏨-01
U[DPT]= 5.671⏨-04
   66    1.340⏨-07    2.882067⏨-01    2.882⏨-01    2.901⏨-01
U[DPT]= 5.506⏨-04
   67    1.350⏨-07    1.994432⏨-01    1.994⏨-01    2.013⏨-01
U[DPT]= 5.346⏨-04
   68    1.360⏨-07    1.231911⏨-01    1.232⏨-01    1.251⏨-01
U[DPT]= 5.192⏨-04
   69    1.370⏨-07    5.850180⏨-02    5.850⏨-02    6.047⏨-02
U[DPT]= 5.044⏨-04
   70    1.380⏨-07    4.489628⏨-03    4.490⏨-03    6.486⏨-03
U[DPT]= 4.901⏨-04
   71    1.390⏨-07   -3.967248⏨-02   -3.967⏨-02   -3.765⏨-02
U[DPT]= 4.763⏨-04
   72    1.400⏨-07   -7.475650⏨-02   -7.476⏨-02   -7.272⏨-02
U[DPT]= 4.630⏨-04
   73    1.410⏨-07   -1.014828⏨-01   -1.015⏨-01   -9.943⏨-02
U[DPT]= 4.501⏨-04
   74    1.420⏨-07   -1.205235⏨-01   -1.205⏨-01   -1.185⏨-01
U[DPT]= 4.377⏨-04
   75    1.430⏨-07   -1.325055⏨-01   -1.325⏨-01   -1.304⏨-01
U[DPT]= 4.257⏨-04
   76    1.440⏨-07   -1.380130⏨-01   -1.380⏨-01   -1.359⏨-01
STEP SIZE INCREASED   H= 2.000⏨-09

U[DPT]= 4.030⏨-04
   77    1.460⏨-07   -1.266656⏨-01   -1.267⏨-01   -1.246⏨-01
U[DPT]= 3.818⏨-04
   78    1.480⏨-07   -9.641882⏨-02   -9.642⏨-02   -9.437⏨-02
U[DPT]= 3.620⏨-04
   79    1.500⏨-07   -5.054402⏨-02   -5.054⏨-02   -4.852⏨-02
U[DPT]= 3.434⏨-04
   80    1.520⏨-07    8.128217⏨-03    8.128⏨-03    1.012⏨-02
U[DPT]= 3.260⏨-04
   81    1.540⏨-07    7.715343⏨-02    7.715⏨-02    7.911⏨-02
U[DPT]= 3.097⏨-04
   82    1.560⏨-07    1.544250⏨-01    1.544⏨-01    1.563⏨-01
U[DPT]= 2.944⏨-04
   83    1.580⏨-07    2.381320⏨-01    2.381⏨-01    2.400⏨-01
U[DPT]= 2.801⏨-04
```

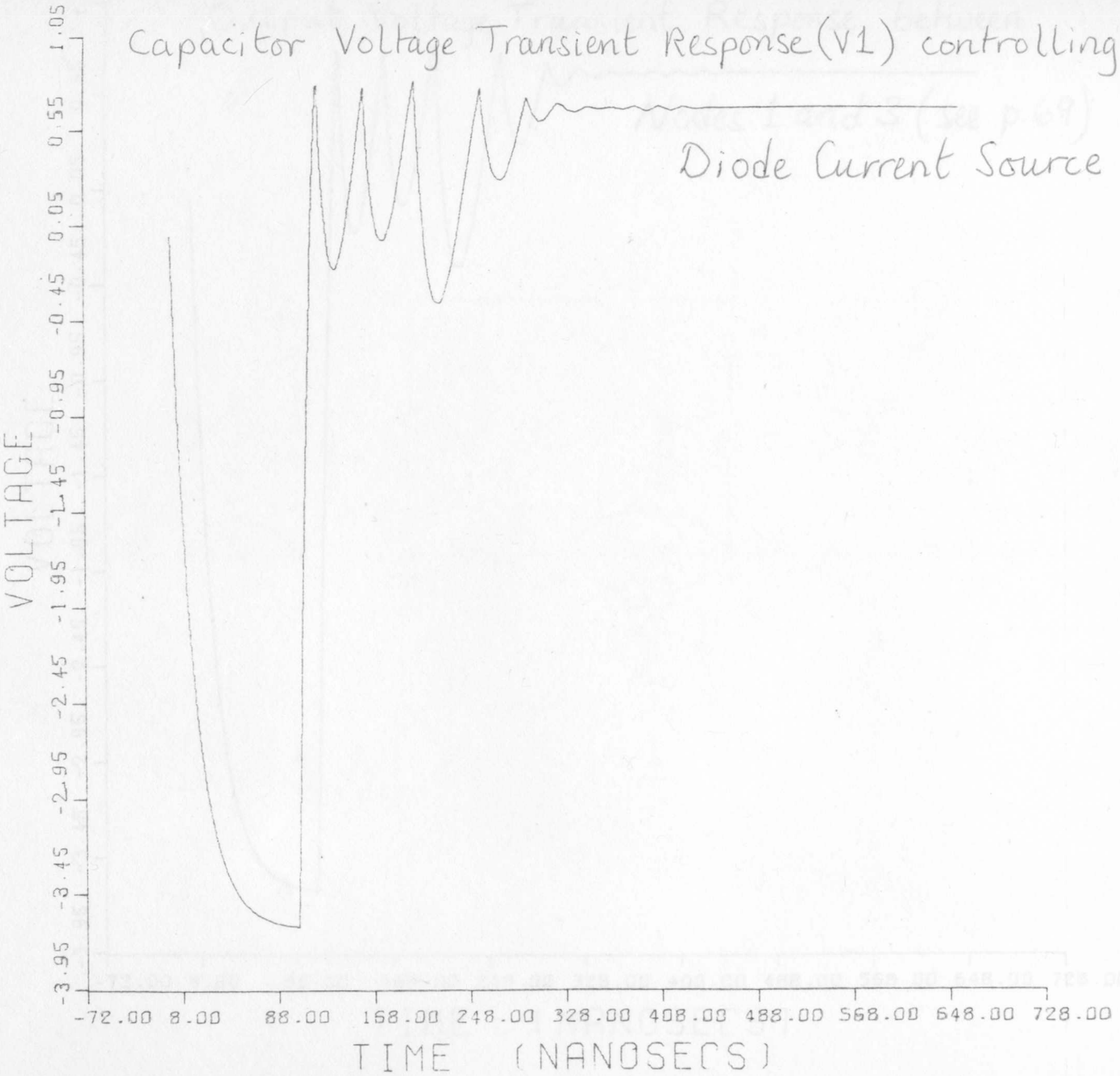
```
      84     1.600₁₀-07     3.267212₁₀-01     3.267₁₀-01     3.286₁₀-01
  U[DPT]= 2.666₁₀-04
      85     1.620₁₀-07     4.188630₁₀-01     4.189₁₀-01     4.207₁₀-01
  U[DPT]= 2.539₁₀-04
      86     1.640₁₀-07     5.134012₁₀-01     5.134₁₀-01     5.151₁₀-01
  STEP SIZE INCREASED     H= 5.000₁₀-09

  U[DPT]= 2.264₁₀-04
      87     1.690₁₀-07     7.689550₁₀-01     7.690₁₀-01     7.706₁₀-01
  U[DPT]= 9.165₁₀-04
  STEP SIZE DECREASED     H= 2.000₁₀-09

  U[DPT]= 5.089₁₀-04
      88     1.760₁₀-07     5.856142₁₀-01     5.856₁₀-01     5.873₁₀-01
  U[DPT]= 4.888₁₀-04
      89     1.780₁₀-07     4.403779₁₀-01     4.404₁₀-01     4.422₁₀-01
  U[DPT]= 4.676₁₀-04
      90     1.800₁₀-07     3.303390₁₀-01     3.303₁₀-01     3.322₁₀-01
  U[DPT]= 4.475₁₀-04
      91     1.820₁₀-07     2.507967₁₀-01     2.508₁₀-01     2.527₁₀-01
  U[DPT]= 4.284₁₀-04
      92     1.840₁₀-07     1.975653₁₀-01     1.976₁₀-01     1.995₁₀-01
  U[DPT]= 4.104₁₀-04
      93     1.860₁₀-07     1.669569₁₀-01     1.670₁₀-01     1.689₁₀-01
  U[DPT]= 3.933₁₀-04
      94     1.880₁₀-07     1.557265₁₀-01     1.557₁₀-01     1.576₁₀-01
  U[DPT]= 3.771₁₀-04
      95     1.900₁₀-07     1.610220₁₀-01     1.610₁₀-01     1.629₁₀-01
  U[DPT]= 3.617₁₀-04
      96     1.920₁₀-07     1.803393₁₀-01     1.803₁₀-01     1.822₁₀-01
  U[DPT]= 3.471₁₀-04
      97     1.940₁₀-07     2.114826₁₀-01     2.115₁₀-01     2.134₁₀-01
  U[DPT]= 3.332₁₀-04
      98     1.960₁₀-07     2.525287₁₀-01     2.525₁₀-01     2.544₁₀-01
  U[DPT]= 3.200₁₀-04
      99     1.980₁₀-07     3.017957₁₀-01     3.018₁₀-01     3.036₁₀-01
  U[DPT]= 3.075₁₀-04
     100     2.000₁₀-07     3.578144₁₀-01     3.578₁₀-01     3.596₁₀-01
  STEP SIZE INCREASED     H= 5.000₁₀-09

  U[DPT]= 2.789₁₀-04
     101     2.050₁₀-07     5.360983₁₀-01     5.361₁₀-01     5.378₁₀-01
  U[DPT]= 2.548₁₀-04
     102     2.100₁₀-07     7.241229₁₀-01     7.241₁₀-01     7.258₁₀-01
  U[DPT]= 4.112₁₀-04
     103     2.150₁₀-07     5.252481₁₀-01     5.252₁₀-01     5.270₁₀-01
  U[DPT]= 3.764₁₀-04
     104     2.200₁₀-07     4.502643₁₀-01     4.503₁₀-01     4.520₁₀-01
```

```
 U[DPT]= 3.445₁₀-04
   105     2.250₁₀-07     4.629992₁₀-01     4.630₁₀-01     4.648₁₀-01
 U[DPT]= 3.159₁₀-04
   106     2.300₁₀-07     5.350099₁₀-01     5.350₁₀-01     5.367₁₀-01
 U[DPT]= 2.913₁₀-04
   107     2.350₁₀-07     6.436219₁₀-01     6.436₁₀-01     6.453₁₀-01
 U[DRT]= 2.872₁₀-04
   108     2.400₁₀-07     7.350141₁₀-01     7.350₁₀-01     7.366₁₀-01
 U[DPT]= 4.753₁₀-04
   109     2.450₁₀-07     3.934013₁₀-01     3.934₁₀-01     3.952₁₀-01
 U[DPT]= 4.388₁₀-04
   110     2.500₁₀-07     2.137064₁₀-01     2.137₁₀-01     2.156₁₀-01
 U[DPT]= 4.057₁₀-04
   111     2.550₁₀-07     1.495429₁₀-01     1.495₁₀-01     1.515₁₀-01
 U[DPT]= 3.756₁₀-04
   112     2.600₁₀-07     1.664503₁₀-01     1.665₁₀-01     1.684₁₀-01
 U[DPT]= 3.483₁₀-04
   113     2.650₁₀-07     2.389797₁₀-01     2.390₁₀-01     2.409₁₀-01
 U[DPT]= 3.234₁₀-04
   114     2.700₁₀-07     3.484362₁₀-01     3.484₁₀-01     3.503₁₀-01
 U[DPT]= 3.008₁₀-04
   115     2.750₁₀-07     4.811649₁₀-01     4.812₁₀-01     4.829₁₀-01
 U[DPT]= 2.803₁₀-04
   116     2.800₁₀-07     6.268191₁₀-01     6.268₁₀-01     6.285₁₀-01
 U[DPT]= 2.717₁₀-04
   117     2.850₁₀-07     7.561278₁₀-01     7.561₁₀-01     7.577₁₀-01
 U[DPT]= 5.662₁₀-04
 STEP SIZE DECREASED    H= 2.000₁₀-09

 U[DPT]= 3.912₁₀-04
   118     2.920₁₀-07     6.855880₁₀-01     6.856₁₀-01     6.872₁₀-01
 U[DPT]= 4.002₁₀-04
   119     2.940₁₀-07     6.138225₁₀-01     6.138₁₀-01     6.155₁₀-01
 U[DPT]= 3.925₁₀-04
   120     2.960₁₀-07     5.568707₁₀-01     5.569₁₀-01     5.586₁₀-01
 U[DPT]= 3.827₁₀-04
   121     2.980₁₀-07     5.150827₁₀-01     5.151₁₀-01     5.168₁₀-01
 U[DPT]= 3.728₁₀-04
   122     3.000₁₀-07     4.869812₁₀-01     4.870₁₀-01     4.887₁₀-01
 U[DPT]= 3.632₁₀-04
   123     3.020₁₀-07     4.709407₁₀-01     4.709₁₀-01     4.727₁₀-01
 U[DPT]= 3.538₁₀-04
   124     3.040₁₀-07     4.654415₁₀-01     4.654₁₀-01     4.672₁₀-01
 U[DPT]= 3.448₁₀-04
   125     3.060₁₀-07     4.691081₁₀-01     4.691₁₀-01     4.709₁₀-01
 U[DPT]= 3.360₁₀-04
   126     3.080₁₀-07     4.807055₁₀-01     4.807₁₀-01     4.825₁₀-01
 U[DPT]= 3.275₁₀-04
```

```
   127      3.100₁₀-07     4.991224₁₀-01     4.991₁₀-01      5.009₁₀-01
STEP SIZE INCREASED        H= 5.000₁₀-09

U[DPT]= 3.077₁₀-04
   128      3.150₁₀-07     5.805323₁₀-01     5.805₁₀-01      5.822₁₀-01
U[DPT]= 2.918₁₀-04
   129      3.200₁₀-07     6.771043₁₀-01     6.771₁₀-01      6.788₁₀-01
U[DPT]= 3.091₁₀-04
   130      3.250₁₀-07     7.125907₁₀-01     7.126₁₀-01      7.142₁₀-01
U[DPT]= 3.783₁₀-04
   131      3.300₁₀-07     5.882924₁₀-01     5.883₁₀-01      5.900₁₀-01
U[DPT]= 3.596₁₀-04
   132      3.350₁₀-07     5.347249₁₀-01     5.347₁₀-01      5.365₁₀-01
U[DPT]= 3.399₁₀-04
   133      3.400₁₀-07     5.372447₁₀-01     5.372₁₀-01      5.390₁₀-01
U[DPT]= 3.215₁₀-04
   134      3.450₁₀-07     5.792150₁₀-01     5.792₁₀-01      5.809₁₀-01
U[DPT]= 3.061₁₀-04
   135      3.500₁₀-07     6.448041₁₀-01     6.448₁₀-01      6.465₁₀-01
U[DPT]= 3.029₁₀-04
   136      3.550₁₀-07     7.015780₁₀-01     7.016₁₀-01      7.032₁₀-01
U[DPT]= 3.466₁₀-04
   137      3.600₁₀-07     6.492658₁₀-01     6.493₁₀-01      6.509₁₀-01
U[DPT]= 3.430₁₀-04
   138      3.650₁₀-07     6.173921₁₀-01     6.174₁₀-01      6.191₁₀-01
U[DPT]= 3.313₁₀-04
   139      3.700₁₀-07     6.187933₁₀-01     6.188₁₀-01      6.205₁₀-01
U[DPT]= 3.206₁₀-04
   140      3.750₁₀-07     6.432949₁₀-01     6.433₁₀-01      6.450₁₀-01
U[DPT]= 3.163₁₀-04
   141      3.800₁₀-07     6.711159₁₀-01     6.711₁₀-01      6.728₁₀-01
STEP SIZE INCREASED        H= 1.000₁₀-08

U[DPT]= 3.340₁₀-04
   142      3.900₁₀-07     6.405777₁₀-01     6.406₁₀-01      6.423₁₀-01
U[DPT]= 3.233₁₀-04
   143      4.000₁₀-07     6.639734₁₀-01     6.640₁₀-01      6.656₁₀-01
U[DPT]= 3.315₁₀-04
   144      4.100₁₀-07     6.459157₁₀-01     6.459₁₀-01      6.476₁₀-01
U[DPT]= 3.246₁₀-04
   145      4.200₁₀-07     6.620568₁₀-01     6.621₁₀-01      6.637₁₀-01
U[DPT]= 3.305₁₀-04
   146      4.300₁₀-07     6.485963₁₀-01     6.486₁₀-01      6.503₁₀-01
U[DPT]= 3.256₁₀-04
   147      4.400₁₀-07     6.596448₁₀-01     6.596₁₀-01      6.613₁₀-01
U[DPT]= 3.291₁₀-04
   148      4.500₁₀-07     6.524479₁₀-01     6.524₁₀-01      6.541₁₀-01
U[DPT]= 3.270₁₀-04
```

```
   149    4.600₁₀-07     6.564874₁₀-01     6.565₁₀-01     6.582₁₀-01
U[DPT]= 3.279₁₀-04
   150    4.700₁₀-07     6.554530₁₀-01     6.555₁₀-01     6.571₁₀-01
U[DPT]= 3.279₁₀-04
   151    4.800₁₀-07     6.546255₁₀-01     6.546₁₀-01     6.563₁₀-01
STEP SIZE INCREASED   H= 2.000₁₀-08

STEP SIZE DECREASED HMAX= 1.000₁₀-08


U[DPT]= 3.274₁₀-04
   152    4.900₁₀-07     6.560817₁₀-01     6.561₁₀-01     6.578₁₀-01
U[DPT]= 3.280₁₀-04
   153    5.000₁₀-07     6.549426₁₀-01     6.549₁₀-01     6.566₁₀-01
U[DPT]= 3.277₁₀-04
   154    5.100₁₀-07     6.553651₁₀-01     6.554₁₀-01     6.570₁₀-01
U[DPT]= 3.277₁₀-04
   155    5.200₁₀-07     6.555529₁₀-01     6.556₁₀-01     6.572₁₀-01
U[DPT]= 3.278₁₀-04
   156    5.300₁₀-07     6.551546₁₀-01     6.552₁₀-01     6.568₁₀-01
U[DPT]= 3.277₁₀-04
   157    5.400₁₀-07     6.554100₁₀-01     6.554₁₀-01     6.571₁₀-01
U[DPT]= 3.277₁₀-04
   158    5.500₁₀-07     6.554048₁₀-01     6.554₁₀-01     6.571₁₀-01
U[DPT]= 3.278₁₀-04
   159    5.600₁₀-07     6.552743₁₀-01     6.553₁₀-01     6.569₁₀-01
U[DPT]= 3.277₁₀-04
   160    5.700₁₀-07     6.553766₁₀-01     6.554₁₀-01     6.570₁₀-01
U[DPT]= 3.277₁₀-04
   161    5.800₁₀-07     6.553717₁₀-01     6.554₁₀-01     6.570₁₀-01
U[DPT]= 3.278₁₀-04
   162    5.900₁₀-07     6.553209₁₀-01     6.553₁₀-01     6.570₁₀-01
U[DPT]= 3.277₁₀-04
   163    6.000₁₀-07     6.553565₁₀-01     6.554₁₀-01     6.570₁₀-01
U[DPT]= 3.277₁₀-04
   164    6.100₁₀-07     6.553616₁₀-01     6.554₁₀-01     6.570₁₀-01
U[DPT]= 3.277₁₀-04
   165    6.200₁₀-07     6.553397₁₀-01     6.553₁₀-01     6.570₁₀-01
U[DPT]= 3.277₁₀-04
   166    6.300₁₀-07     6.553490₁₀-01     6.553₁₀-01     6.570₁₀-01
U[DPT]= 3.277₁₀-04
   167    6.400₁₀-07     6.553559₁₀-01     6.554₁₀-01     6.570₁₀-01
U[DPT]= 3.277₁₀-04
NO MORE TRANSIENTS AND STEADY STATE REACHED
```

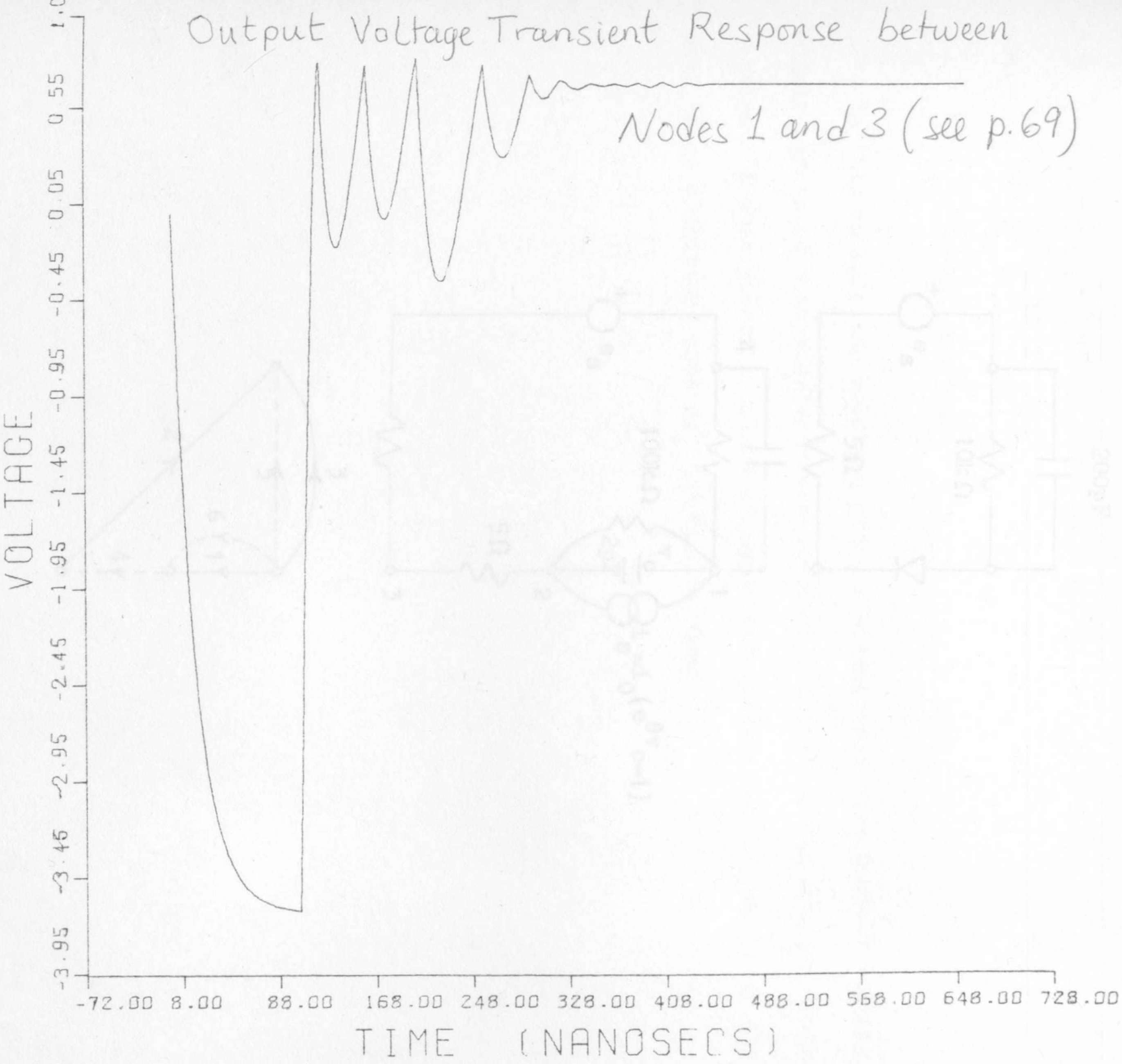
Non-Linear Model of Diode



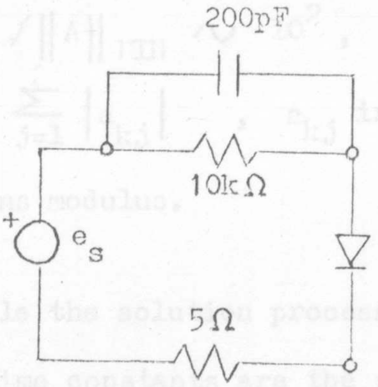
Capacitor Voltage Transient Response (V1) controlling Diode Current Source

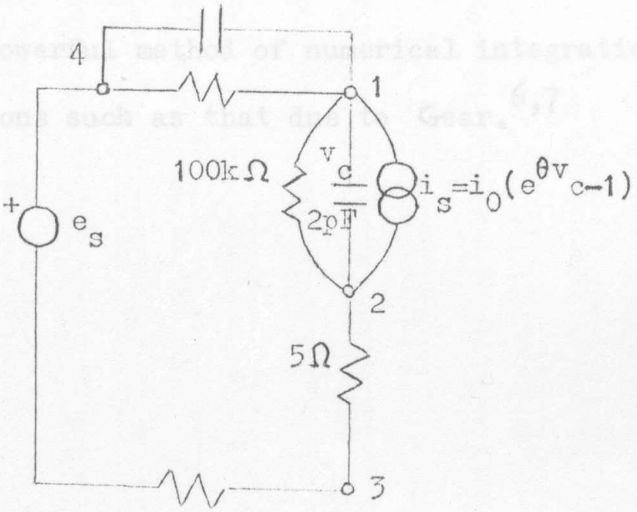
Non-Linear Model of Diode

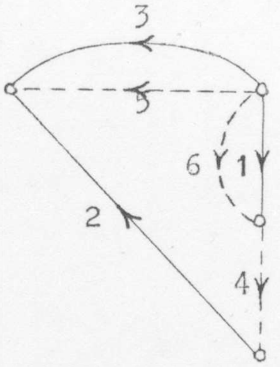
Output Voltage Transient Response between Nodes 1 and 3 (see p.69)



Ex. 11. Exponential Model of Diode







In this problem introducing a second capacitor in combination with the 10k resistor slows the solution process down drastically. The time constants for the two capacitors are approximately separated by 10^2, as measured by the ratio of the maximum to minimum row norms of the state matrix A,

e.g. $\|A\|_{MAX} / \|A\|_{MIN} \sim 10^2$,

where $\|A\| = \sum_{j=1}^{2} |a_{kj}|$, a_{kj} is the jth element in the kth row,

and $|\;|$ means modulus.

In this example the solution process breaks down immediately showing widely separated time constants are the severest test so far applied to this routine. A more powerful method of numerical integration is needed to deal with these conditions such as that due to Gear.[6,7]

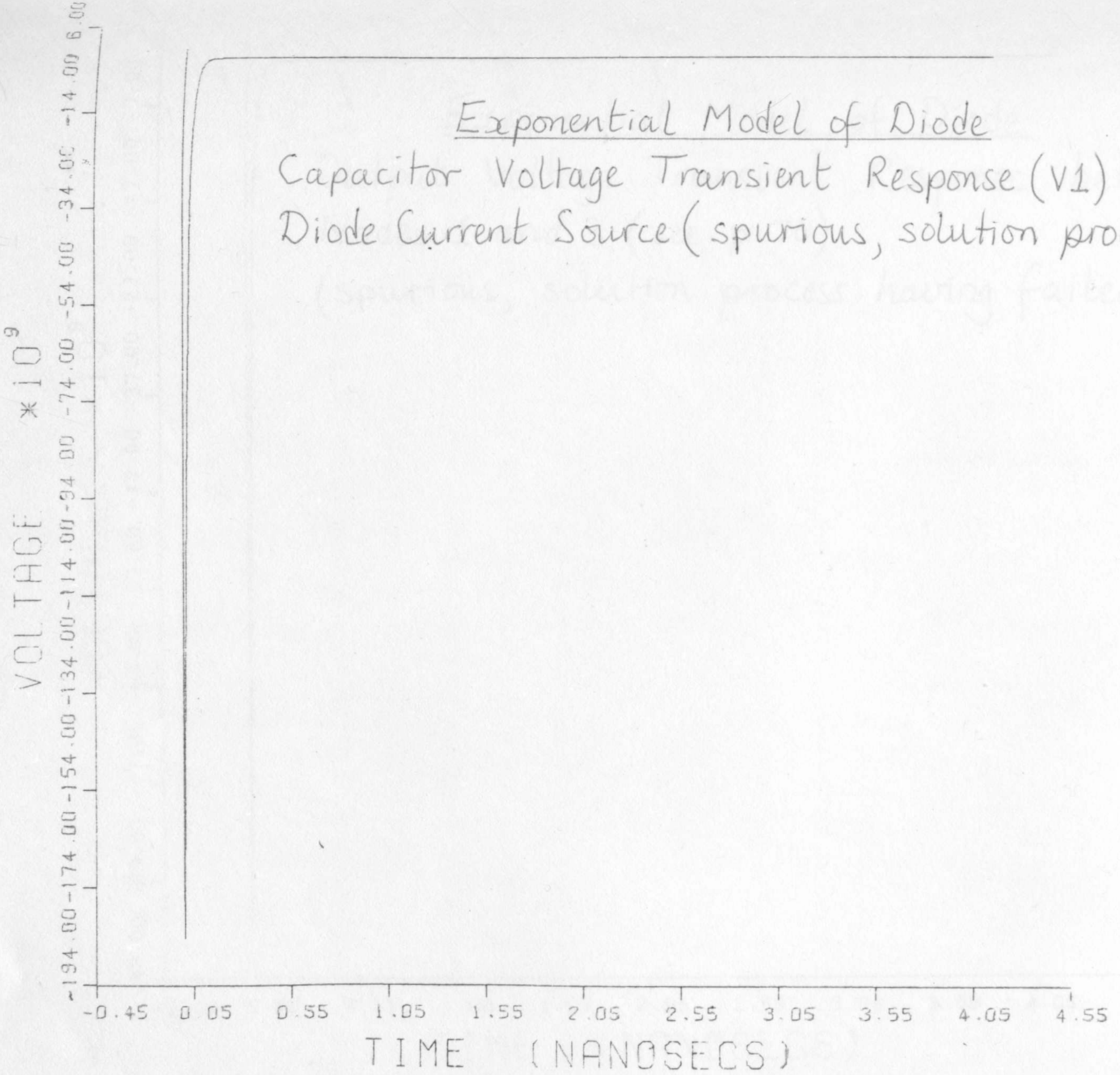
```
EXPONENTIAL  MODEL  OF  DIODE
FOFSVM5 OUTPUT
NRUN=    2

INPUT FOFSVM5;

B=      6   RSOURCE=      2    NY=      2
   B    MN1[B]  MN2[B]  MCOMP[B] ACOMP[B]

   1       2        1        1     2.000⏨-12
   2       3        2        2     5.000⏨+00
   3       3        4        2     5.000⏨+00
   4       1        4        1     2.000⏨-10
   5       1        4        2     1.000⏨+04
   6       2        1        2     1.000⏨+05
    SMN1    SMN2  SMCOMP    SCOL   MSOURCE
        1        2        0        6        2
        3        4        1        3        0

N=    3   L=    3   R=    4   C=    2


OUTPUT FOFSVM5;
ORDER=    2   LC=    0   LR=    1   LL=    0   NC=    0   NR=    1   NL=    0

BETAF[L,N]=
     1     1    -1
     0     0    -1
    -1     0     0

OUTPUT  FOFSVM5;


TR1[N]=         1        3        4

TL1[L]=         2        5        6
NETWORK  BRANCHES  REORDERED  INTO  TREE  BRANCHES  AND  LINKS


        B   MN1[B] MN2[B] MCOMP[B] ACOMP[B]
        1       1        2        1     2.000⏨-12    V[    1]
        2       3        4       52     5.000⏨+00    V[    2]
        3       1        4        2     2.000⏨-10    V[    3]
        4       2        3       51     5.000⏨+00    J[    4]
        5       1        4       53     1.000⏨+04    J[    5]
        6       1        2       54     1.000⏨+05    J[    6]

     SMN1   SMN2   SMCOMP   SCOL   MSOURCE
```

```
       1        2        0        6        2
       3        4        1        2        0

TB3=
      2       4       6
TL3=
      1       3       5

BETAF[L,N]=
     0    0   -1
   -1   -1   -1
   -1   -1   -1
TB4=
      1       2       3
TL4=
      4       5       6

BETAF[L,N]=
     1    1   -1
     0    0   -1
   -1    0    0
SMALLEST OVER LARGEST PIVOT= 1.000⏨+00
SMALLEST OVER LARGEST PIVOT= 1.000⏨+00
SMALLEST OVER LARGEST PIVOT= 5.000⏨-04
THE  STATE  VARIABLES  ARE
(   V[   1]   V[   3])

A=
   -5.001⏨+10     5.000⏨+10

    5.000⏨+08    -5.005⏨+08


B=
   -5.000⏨+11     5.000⏨+10

    0.000⏨+00    -5.000⏨+08


C=
    1.000⏨+00     0.000⏨+00

    5.000⏨-01     5.000⏨-01


D=
    0.000⏨+00     0.000⏨+00
    0.000⏨+00    5.000⏨-01


  WHERE  DX/DT=AX+BU    AND   Y=CX+DU

  H= 9.000⏨-13
  NINT=     2
  IS= 5.000⏨-11  THETA= 2.641⏨+01
  TINT= 0.000⏨+00   UINT=-4.000⏨+00
  TINT= 1.000⏨-07   UINT=-4.000⏨+00
  TINT= 1.000⏨-07   UINT= 4.000⏨+00
  TINT= 1.000⏨+06   UINT= 4.000⏨+00
  EPS= 1.000⏨-06  EPS1= 2.500⏨-02  EF
  T0= 0.000⏨+00  TN= 1.000⏨-06  NPTS=

     F=0  T= 0.000⏨+00     0.000⏨+00
  VMAX= 4.000⏨+00    IMAX= 8.000⏨-01

  U[EXP]= 3.841⏨+11
       1    9.000⏨-13    -1.690298⏨+11
  U[EXP]= 5.199⏨+10
       2    1.800⏨-12    -1.844704⏨+11
  U[EXP]= 1.370⏨+10
       3    2.700⏨-12    -1.823894⏨+11
  U[EXP]= 5.041⏨+09
       4    3.600⏨-12    -1.765919⏨+11
  U[EXP]= 2.265⏨+09
       5    4.500⏨-12    -1.698315⏨+11
  U[EXP]= 1.163⏨+09
       6    5.400⏨-12    -1.628870⏨+11
  U[EXP]= 6.568⏨+08
       7    6.300⏨-12    -1.560285⏨+11
  U[EXP]= 3.983⏨+08
       8    7.200⏨-12    -1.493612⏨+11
  U[EXP]= 2.554⏨+08
       9    8.100⏨-12    -1.429274⏨+11
  U[EXP]= 1.712⏨+08
      10    9.000⏨-12    -1.367425⏨+11
  U[EXP]= 1.190⏨+08
      11    9.900⏨-12    -1.308094⏨+11
  U[EXP]= 8.528⏨+07
      12    1.080⏨-11    -1.251252⏨+11
  U[EXP]= 6.269⏨+07
      13    1.170⏨-11    -1.196837⏨+11
  U[EXP]= 4.711⏨+07
      14    1.260⏨-11    -1.144771⏨+11
  U[EXP]= 3.608⏨+07
```

```
PS2= 1.000₁₀-01
=   .  500

 0.000₁₀+00     0.000₁₀+00     0.000₁₀+00

  -3.831113₁₀+07   -1.690₁₀+11   -8.453₁₀+10

  -1.178397₁₀+08   -1.845₁₀+11   -9.229₁₀+10

  -2.003078₁₀+08   -1.824₁₀+11   -9.129₁₀+10

  -2.809602₁₀+08   -1.766₁₀+11   -8.844₁₀+10

  -3.587498₁₀+08   -1.698₁₀+11   -8.510₁₀+10

  -4.334210₁₀+08   -1.629₁₀+11   -8.166₁₀+10

  -5.049539₁₀+08   -1.560₁₀+11   -7.827₁₀+10

  -5.734122₁₀+08   -1.494₁₀+11   -7.497₁₀+10

  -6.388930₁₀+08   -1.429₁₀+11   -7.178₁₀+10

  -7.015062₁₀+08   -1.367₁₀+11   -6.872₁₀+10

  -7.613657₁₀+08   -1.308₁₀+11   -6.579₁₀+10

  -8.185854₁₀+08   -1.251₁₀+11   -6.297₁₀+10

  -8.732769₁₀+08   -1.197₁₀+11   -6.028₁₀+10

  -9.255490₁₀+08   -1.145₁₀+11   -5.770₁₀+10
```

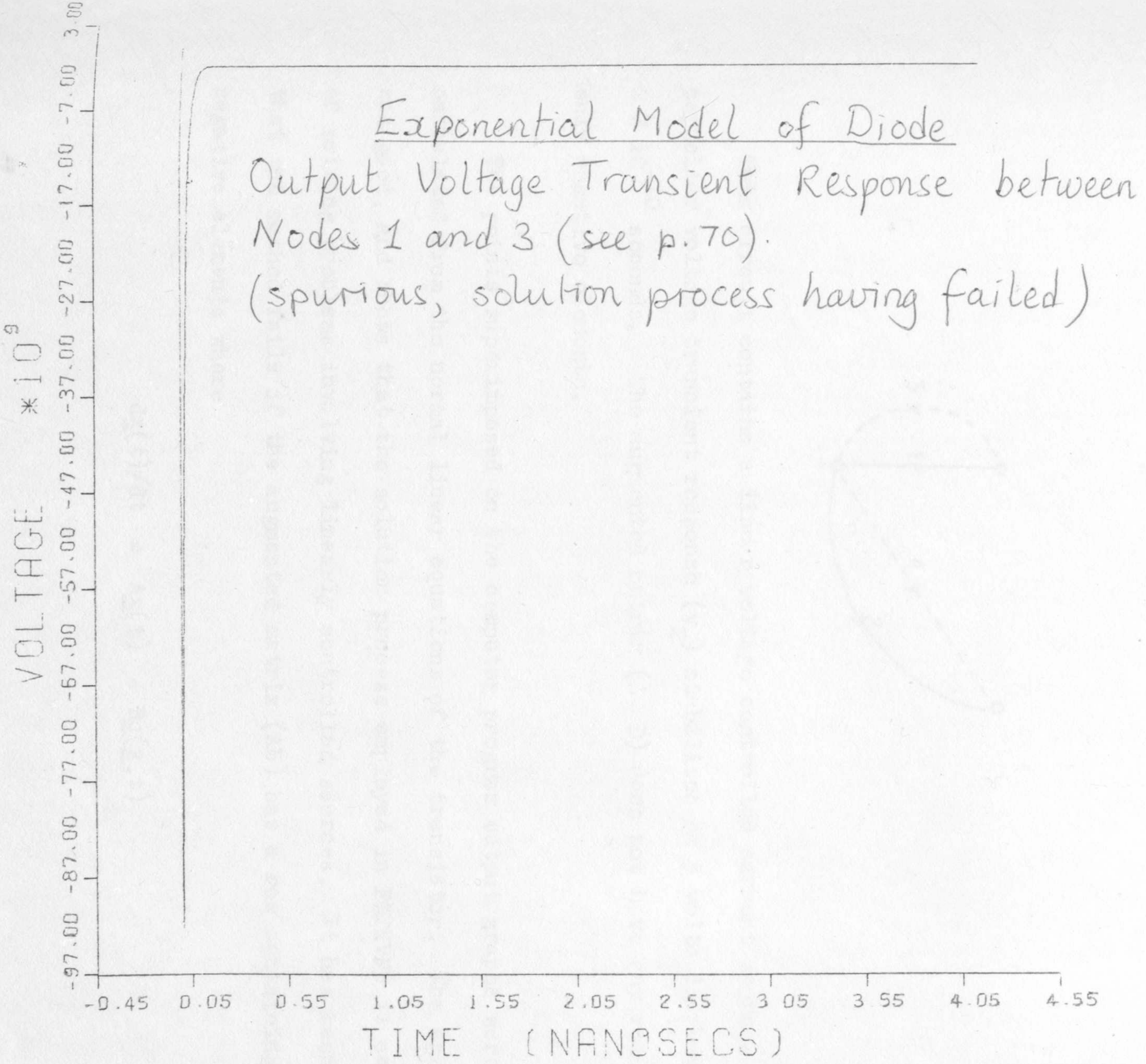
```
     490     4.061₁₀-09   -2.045935₁₀+09    -2.046117₁₀+09    -2.046₁₀+09
   U[EXP]= 6.591₁₀-03
     491     4.070₁₀-09   -2.045925₁₀+09    -2.046107₁₀+09    -2.046₁₀+09
   U[EXP]= 6.534₁₀-03
     492     4.079₁₀-09   -2.045915₁₀+09    -2.046097₁₀+09    -2.046₁₀+09
   U[EXP]= 6.476₁₀-03
     493     4.088₁₀-09   -2.045905₁₀+09    -2.046087₁₀+09    -2.046₁₀+09
   U[EXP]= 6.420₁₀-03
     494     4.097₁₀-09   -2.045895₁₀+09    -2.046077₁₀+09    -2.046₁₀+09
   U[EXP]= 6.364₁₀-03
     495     4.106₁₀-09   -2.045885₁₀+09    -2.046067₁₀+09    -2.046₁₀+09
   U[EXP]= 6.308₁₀-03
     496     4.115₁₀-09   -2.045874₁₀+09    -2.046057₁₀+09    -2.046₁₀+09
   U[EXP]= 6.253₁₀-03
     497     4.124₁₀-09   -2.045864₁₀+09    -2.046047₁₀+09    -2.046₁₀+09
   U[EXP]= 6.199₁₀-03
     498     4.133₁₀-09   -2.045854₁₀+09    -2.046037₁₀+09    -2.046₁₀+09
   U[EXP]= 6.145₁₀-03
     499     4.142₁₀-09   -2.045844₁₀+09    -2.046027₁₀+09    -2.046₁₀+09
   U[EXP]= 6.092₁₀-03
     500     4.151₁₀-09   -2.045834₁₀+09    -2.046017₁₀+09    -2.046₁₀+09
   T=-1.000₁₀+00  NPTS=  501
```

```
-2.046⏨+09

-2.046⏨+09

-2.046⏨+09

-2.046⏨+09

-2.046⏨+09

-2.046⏨+09

-2.046⏨+09

-2.046⏨+09

-2.046⏨+09

-2.046⏨+09

-2.046⏨+09
```



Exponential Model of Diode

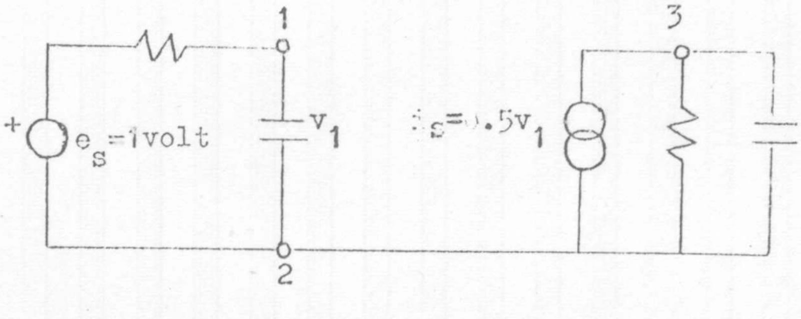
Capacitor Voltage Transient Response (V1) controlling Diode Current Source (spurious, solution process having failed)

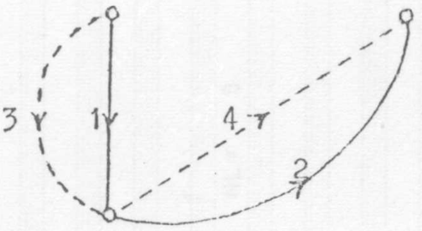


Exponential Model of Diode

Output Voltage Transient Response between Nodes 1 and 3 (see p. 70).
(spurious, solution process having failed)

Ex. 12. Linear Model of Transistor





This circuit contains a linear voltage controlled current source. The capacitor voltage transient response (v_2) stabilises at 5 volts in about 2 X 10^{-10} seconds. The augmented matrix (A B) does not have any rows containi only negative elements.

The points superimposed on the computer program output graphs were calculated from the normal linear equations of the transistor. The agreement was good, and shows that the solution process employed in FOFSVM5 is capable of solving problems involving linearly controlled sources. It has been found that the method fails if the augmented matrix (AB) has a row containing all negative elements where

$$d\underline{x}(t)/dt = A\underline{x}(t) + B\underline{u}(\underline{x},t)$$

```
LINEAR  MODEL  OF  TRANSISTOR
FOFSVM5 OUTPUT
NRUN=    5

INPUT FOFSVM5;

B=     4   RSOURCE=     2    NY=     2
    B    MN1[B]  MN2[B]   MCOMP[B] ACOMP[B]

    1       1        2        1      1.000⏨-12
    2       1        2        2      1.000⏨+01
    3       2        3        2      1.000⏨+01
    4       2        3        1      1.000⏨-12
     SMN1    SMN2  SMCOMP    SCOL   MSOURCE
         3        2        0        3        1
         2        1        1        2        0

N=   2   L=   2   R=   2   C=   2


OUTPUT FOFSVM5;
ORDER=   2   LC=    0   LR=    0   LL=    0   NC=    0   NR=    0   NL=    0

BETAF[L,N]=
   -1     0
    0    -1

OUTPUT  FOFSVM5;


TB1[N]=         1         4


IL1[L]=         2         3
NETWORK  BRANCHES  REORDERED  INTO  TREE  BRANCHES  AND  LINKS


      B  MN1[B] MN2[B] MCOMP[B] ACOMP[B]
      1      1      2       1    1.000⏨-12   V[    1]
      2      2      3       2    1.000⏨-12   V[    2]
      3      1      2      51    1.000⏨+01   J[    3]
      4      2      3      52    1.000⏨+01   J[    4]

     SMN1   SMN2   SMCOMP   SCOL   MSOURCE
        3       2       0       4       1
        2       1       1       3       0

SMALLEST OVER LARGEST PIVOT= 1.000⏨+00
THE  STATE  VARIABLES  ARE
```

```
(   V[   1]   V[   2])

A=
   -1.000₁₀+11    0.000₁₀+00

    0.000₁₀+00   -1.000₁₀+11

B=
    0.000₁₀+00    1.000₁₀+11

    1.000₁₀+12    0.000₁₀+00

C=
   -1.000₁₀+00    0.000₁₀+00

    0.000₁₀+00   -1.000₁₀+00

D=
    0.000₁₀+00    0.000₁₀+00

    0.000₁₀+00    0.000₁₀+00

WHERE  DX/DT=AX+BU    AND  Y=CX+DU

H= 9.000₁₀-13
NINT=     1
IS= 5.000₁₀-01
TINT= 0.000₁₀+00    UINT= 1.000₁₀+00
TINT= 1.000₁₀+06    UINT= 1.000₁₀+00
EPS= 1.000₁₀-06  EPS1= 2.500₁₀-02  EPS2= 1.000₁₀-01
T0= 0.000₁₀+00  TN= 3.000₁₀-08  NPTS=      500

  F=0  T= 0.000₁₀+00     0.000₁₀+00     0.000₁₀+00     0.000₁₀+00     0.000₁₀+00
VMAX= 1.000₁₀+00    IMAX= 1.000₁₀-01

U[DPT]= 1.832₁₀-02
    1    9.000₁₀-13     8.606882₁₀-02    1.576405₁₀-02    -8.607₁₀-02    -1.576₁₀-02
U[DPT]=-3.473₁₀-02
    2    1.80)₁₀-12     1.647298₁₀-01   -1.548580₁₀-02    -1.647₁₀-01     1.549₁₀-02
U[DPT]=-6.981₁₀-02
    3    2.700₁₀-12     2.366205₁₀-01   -7.423664₁₀-02    -2.366₁₀-01     7.424₁₀-02
U[DPT]=-1.005₁₀-01
    4    3.600₁₀-12     3.023237₁₀-01   -1.543185₁₀-01    -3.023₁₀-01     1.543₁₀-01
   U[DPT]=-1.284₁₀-01
         5    4.500₁₀-12      3.623719₁₀-01
   U[DPT]=-1.541₁₀-01
         6    5.400₁₀-12      4.172518₁₀-01
   U[DPT]=-1.778₁₀-01
         7    6.300₁₀-12      4.674083₁₀-01
   U[DPT]=-1.998₁₀-01
         8    7.200₁₀-12      5.132478₁₀-01
   U[DPT]=-2.202₁₀-01
         9    8.100₁₀-12      5.551420₁₀-01
   U[DPT]=-2.391₁₀-01
        10    9.000₁₀-12      5.934305₁₀-01
   U[DPT]=-2.567₁₀-01
   STEP SIZE DECREASED        H= 4.500₁₀-13

   U[DPT]=-2.491₁₀-01
        11    9.450₁₀-12      6.113204₁₀-01
   U[DPT]=-2.585₁₀-01
        12    9.900₁₀-12      6.284232₁₀-01
   U[DPT]=-2.674₁₀-01
        13    1.035₁₀-11      6.447734₁₀-01
   U[DPT]=-2.758₁₀-01
        14    1.080₁₀-11      6.604042₁₀-01
   U[DPT]=-2.839₁₀-01
        15    1.125₁₀-11      6.753471₁₀-01
   U[DPT]=-2.915₁₀-01
        16    1.170₁₀-11      6.896326₁₀-01
   U[DPT]=-2.989₁₀-01
        17    1.215₁₀-11      7.032894₁₀-01
   U[DPT]=-3.059₁₀-01
        18    1.260₁₀-11      7.163453₁₀-01
   U[DPT]=-3.126₁₀-01
        19    1.305₁₀-11      7.288267₁₀-01
   U[DPT]=-3.191₁₀-01
        20    1.350₁₀-11      7.407589₁₀-01
   U[DPT]=-3.253₁₀-01
        21    1.395₁₀-11      7.521660₁₀-01
   U[DPT]=-3.313₁₀-01
        22    1.440₁₀-11      7.630712₁₀-01
   U[DPT]=-3.370₁₀-01
        23    1.485₁₀-11      7.734965₁₀-01
   U[DPT]=-3.425₁₀-01
        24    1.530₁₀-11      7.834631₁₀-01
   U[DPT]=-3.478₁₀-01
        25    1.575₁₀-11      7.929912₁₀-01
   U[DPT]=-3.530₁₀-01
        26    1.620₁₀-11      8.020999₁₀-01
   U[DPT]=-3.579₁₀-01
```

```
-2.515346⏨-01   -3.624⏨-01    2.515⏨-01

-3.624969⏨-01   -4.173⏨-01    3.625⏨-01

-4.843454⏨-01   -4.674⏨-01    4.843⏨-01

-6.146318⏨-01   -5.132⏨-01    6.146⏨-01

-7.512507⏨-01   -5.551⏨-01    7.513⏨-01

-8.923915⏨-01   -5.934⏨-01    8.924⏨-01




-9.627348⏨-01   -6.113⏨-01    9.627⏨-01

-1.034120⏨+00   -6.284⏨-01    1.034⏨+00

-1.106278⏨+00   -6.448⏨-01    1.106⏨+00

-1.178974⏨+00   -6.604⏨-01    1.179⏨+00

-1.252009⏨+00   -6.753⏨-01    1.252⏨+00

-1.325206⏨+00   -6.896⏨-01    1.325⏨+00

-1.398409⏨+00   -7.033⏨-01    1.398⏨+00

-1.471482⏨+00   -7.163⏨-01    1.471⏨+00

-1.544303⏨+00   -7.288⏨-01    1.544⏨+00

-1.616763⏨+00   -7.408⏨-01    1.617⏨+00

-1.688766⏨+00   -7.522⏨-01    1.689⏨+00

-1.760225⏨+00   -7.631⏨-01    1.760⏨+00

-1.831065⏨+00   -7.735⏨-01    1.831⏨+00

-1.901217⏨+00   -7.835⏨-01    1.901⏨+00

-1.970620⏨+00   -7.930⏨-01    1.971⏨+00

-2.039220⏨+00   -8.021⏨-01    2.039⏨+00
```

```
    27      1.665₁₀-11      8.108079₁₀-01
U[DPT]=-3.626₁₀-01
    28      1.710₁₀-11      8.191327₁₀-01
U[DPT]=-3.672₁₀-01
    29      1.755₁₀-11      8.270912₁₀-01
U[DPT]=-3.716₁₀-01
    30      1.800₁₀-11      8.346994₁₀-01
U[DPT]=-3.759₁₀-01
    31      1.845₁₀-11      8.419729₁₀-01
U[DPT]=-3.800₁₀-01
    32      1.890₁₀-11      8.489264₁₀-01
U[DPT]=-3.839₁₀-01
    33      1.935₁₀-11      8.555739₁₀-01
U[DPT]=-3.877₁₀-01
    34      1.980₁₀-11      8.619288₁₀-01
U[DPT]=-3.914₁₀-01
    35      2.025₁₀-11      8.680042₁₀-01
U[DPT]=-3.950₁₀-01
    36      2.070₁₀-11      8.738122₁₀-01
U[DPT]=-3.984₁₀-01
    37      2.115₁₀-11      8.793646₁₀-01
U[DPT]=-4.017₁₀-01
    38      2.160₁₀-11      8.846727₁₀-01
U[DPT]=-4.049₁₀-01
    39      2.205₁₀-11      8.897472₁₀-01
U[DPT]=-4.079₁₀-01
    40      2.250₁₀-11      8.945985₁₀-01
U[DPT]=-4.109₁₀-01
    41      2.295₁₀-11      8.992363₁₀-01
U[DPT]=-4.138₁₀-01
    42      2.340₁₀-11      9.036700₁₀-01
U[DPT]=-4.165₁₀-01
    43      2.385₁₀-11      9.079086₁₀-01
STEP SIZE INCREASED     H= 9.000₁₀-13

U[DPT]=-4.216₁₀-01
    44      2.475₁₀-11      9.158348₁₀-01
U[DPT]=-4.264₁₀-01
    45      2.565₁₀-11      9.230789₁₀-01
U[DPT]=-4.308₁₀-01
    46      2.655₁₀-11      9.296994₁₀-01
U[DPT]=-4.350₁₀-01
    47      2.745₁₀-11      9.357502₁₀-01
U[DPT]=-4.389₁₀-01
    48      2.835₁₀-11      9.412801₁₀-01
U[DPT]=-4.426₁₀-01
    49      2.925₁₀-11      9.463341₁₀-01
U[DPT]=-4.460₁₀-01
```

```
 -2.106969₁₀+00   -8.108₁₀-01    2.107₁₀+00

 -2.173826₁₀+00   -8.191₁₀-01    2.174₁₀+00

 -2.239755₁₀+00   -8.271₁₀-01    2.240₁₀+00

 -2.304722₁₀+00   -8.347₁₀-01    2.305₁₀+00

 -2.368701₁₀+00   -8.420₁₀-01    2.369₁₀+00

 -2.431669₁₀+00   -8.489₁₀-01    2.432₁₀+00

 -2.493604₁₀+00   -8.556₁₀-01    2.494₁₀+00

 -2.554492₁₀+00   -8.619₁₀-01    2.554₁₀+00

 -2.614318₁₀+00   -8.680₁₀-01    2.614₁₀+00

 -2.673073₁₀+00   -8.738₁₀-01    2.673₁₀+00

 -2.730748₁₀+00   -8.794₁₀-01    2.731₁₀+00

 -2.787338₁₀+00   -8.847₁₀-01    2.787₁₀+00

 -2.842841₁₀+00   -8.897₁₀-01    2.843₁₀+00

 -2.897254₁₀+00   -8.946₁₀-01    2.897₁₀+00

 -2.950580₁₀+00   -8.992₁₀-01    2.951₁₀+00

 -3.002820₁₀+00   -9.037₁₀-01    3.003₁₀+00

 -3.053978₁₀+00   -9.079₁₀-01    3.054₁₀+00



 -3.154001₁₀+00   -9.158₁₀-01    3.154₁₀+00

 -3.249507₁₀+00   -9.231₁₀-01    3.250₁₀+00

 -3.340631₁₀+00   -9.297₁₀-01    3.341₁₀+00

 -3.427512₁₀+00   -9.358₁₀-01    3.428₁₀+00

 -3.510289₁₀+00   -9.413₁₀-01    3.510₁₀+00

 -3.589103₁₀+00   -9.463₁₀-01    3.589₁₀+00
```

```
      50     3.015₁₀-11     9.509531₁₀-01
  U[DPT]=-4.493₁₀-01
      51     3.105₁₀-11     9.551746₁₀-01
  U[DPT]=-4.523₁₀-01
      52     3.195₁₀-11     9.590327₁₀-01
  U[DPT]=-4.551₁₀-01
      53     3.285₁₀-11     9.625587₁₀-01
  U[DPT]=-4.578₁₀-01
      54     3.375₁₀-11     9.657813₁₀-01
  U[DPT]=-4.602₁₀-01
      55     3.465₁₀-11     9.687265₁₀-01
  U[DPT]=-4.626₁₀-01
      56     3.555₁₀-11     9.714182₁₀-01
  U[DPT]=-4.647₁₀-01
      57     3.645₁₀-11     9.738783₁₀-01
  STEP SIZE INCREASED     H= 1.800₁₀-12

  U[DPT]=-4.685₁₀-01
      58     3.825₁₀-11     9.781813₁₀-01
  U[DPT]=-4.719₁₀-01
      59     4.005₁₀-11     9.817754₁₀-01
  U[DPT]=-4.749₁₀-01
      60     4.185₁₀-11     9.847775₁₀-01
  U[DPT]=-4.776₁₀-01
      61     4.365₁₀-11     9.872851₁₀-01
  U[DPT]=-4.799₁₀-01
      62     4.545₁₀-11     9.893795₁₀-01
  U[DPT]=-4.820₁₀-01
      63     4.725₁₀-11     9.911290₁₀-01
  U[DPT]=-4.839₁₀-01
      64     4.905₁₀-11     9.925903₁₀-01
  U[DPT]=-4.855₁₀-01
      65     5.085₁₀-11     9.938108₁₀-01
  U[DPT]=-4.870₁₀-01
      66     5.265₁₀-11     9.948303₁₀-01
  U[DPT]=-4.883₁₀-01
      67     5.445₁₀-11     9.956819₁₀-01
  STEP SIZE INCREASED     H= 4.500₁₀-12

  U[DPT]=-4.908₁₀-01
      68     5.895₁₀-11     9.972466₁₀-01
  U[DPT]=-4.927₁₀-01
      69     6.345₁₀-11     9.982443₁₀-01
  U[DPT]=-4.942₁₀-01
      70     6.795₁₀-11     9.988805₁₀-01
  U[DPT]=-4.954₁₀-01
      71     7.245₁₀-11     9.992861₁₀-01
  U[DPT]=-4.963₁₀-01
```

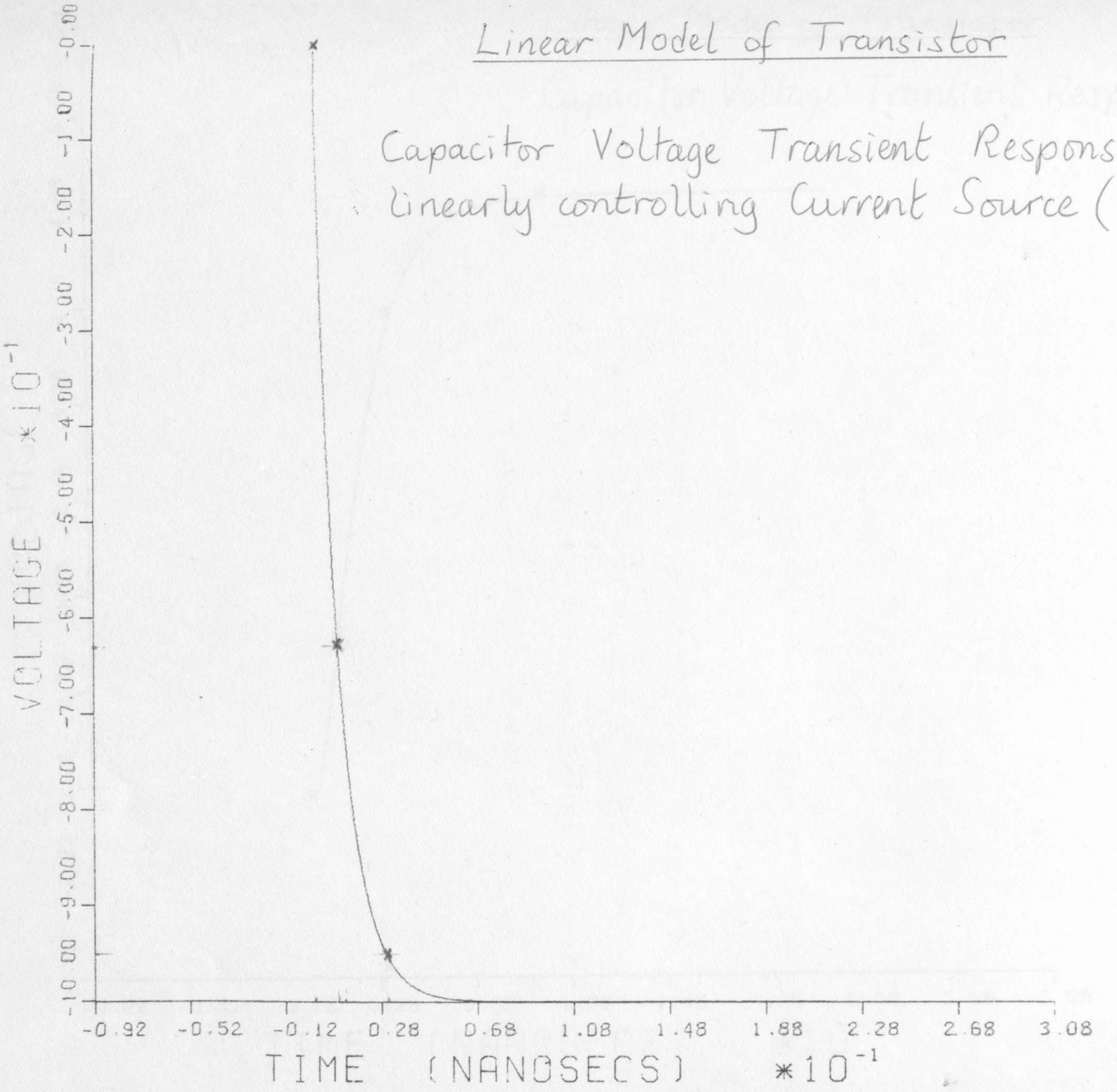
```
-3.664094₁₀+00    -9.510₁₀-01    3.664₁₀+00

-3.735404₁₀+00    -9.552₁₀-01    3.735₁₀+00

-3.803175₁₀+00    -9.590₁₀-01    3.803₁₀+00

-3.867546₁₀+00    -9.626₁₀-01    3.868₁₀+00

-3.928655₁₀+00    -9.658₁₀-01    3.929₁₀+00

-3.986640₁₀+00    -9.687₁₀-01    3.987₁₀+00

-4.041634₁₀+00    -9.714₁₀-01    4.042₁₀+00

-4.093769₁₀+00    -9.739₁₀-01    4.094₁₀+00



-4.191238₁₀+00    -9.782₁₀-01    4.191₁₀+00

-4.278219₁₀+00    -9.818₁₀-01    4.278₁₀+00

-4.355809₁₀+00    -9.848₁₀-01    4.356₁₀+00

-4.424993₁₀+00    -9.873₁₀-01    4.425₁₀+00

-4.486656₁₀+00    -9.894₁₀-01    4.487₁₀+00

-4.541598₁₀+00    -9.911₁₀-01    4.542₁₀+00

-4.590536₁₀+00    -9.926₁₀-01    4.591₁₀+00

-4.634115₁₀+00    -9.938₁₀-01    4.634₁₀+00

-4.672914₁₀+00    -9.948₁₀-01    4.673₁₀+00

-4.707455₁₀+00    -9.957₁₀-01    4.707₁₀+00



-4.780056₁₀+00    -9.972₁₀-01    4.780₁₀+00

-4.833369₁₀+00    -9.982₁₀-01    4.833₁₀+00

-4.872763₁₀+00    -9.989₁₀-01    4.873₁₀+00

-4.902053₁₀+00    -9.993₁₀-01    4.902₁₀+00
```

```
      72      7.695₁₀-11        9.995448₁₀-01
  U[DPT]=-4.970₁₀-01
      73      8.145₁₀-11        9.997097₁₀-01
  U[DPT]=-4.975₁₀-01
      74      8.595₁₀-11        9.998148₁₀-01
  U[DPT]=-4.979₁₀-01
      75      9.045₁₀-11        9.998819₁₀-01
  U[DPT]=-4.983₁₀-01
      76      9.495₁₀-11        9.999247₁₀-01
  U[DPT]=-4.986₁₀-01
      77      9.945₁₀-11        9.999519₁₀-01
  STEP SIZE INCREASED     H= 9.000₁₀-12

  STEP SIZE DECREASED HMAX= 4.500₁₀-12

  U[DPT]=-4.988₁₀-01
      78      1.039₁₀-10        9.999693₁₀-01
  U[DPT]=-4.990₁₀-01
      79      1.084₁₀-10        9.999804₁₀-01
  U[DPT]=-4.991₁₀-01
      80      1.129₁₀-10        9.999874₁₀-01
  U[DPT]=-4.992₁₀-01
      81      1.174₁₀-10        9.999920₁₀-01
  U[DPT]=-4.993₁₀-01
      82      1.219₁₀-10        9.999948₁₀-01
  U[DPT]=-4.994₁₀-01
      83      1.264₁₀-10        9.999967₁₀-01
  U[DPT]=-4.995₁₀-01
      84      1.310₁₀-10        9.999978₁₀-01
  U[DPT]=-4.996₁₀-01
      85      1.355₁₀-10        9.999986₁₀-01
  U[DPT]=-4.996₁₀-01
      86      1.400₁₀-10        9.999990₁₀-01
  U[DPT]=-4.997₁₀-01
      87      1.445₁₀-10        9.999994₁₀-01
  U[DPT]=-4.997₁₀-01
      88      1.490₁₀-10        9.999995₁₀-01
  U[DPT]=-4.997₁₀-01
      89      1.535₁₀-10        9.999997₁₀-01
  U[DPT]=-4.998₁₀-01
      90      1.580₁₀-10        9.999997₁₀-01
  U[DPT]=-4.998₁₀-01
      91      1.625₁₀-10        9.999998₁₀-01
  U[DPT]=-4.998₁₀-01
      92      1.670₁₀-10        9.999998₁₀-01
  U[DPT]=-4.998₁₀-01
      93      1.715₁₀-10        9.999998₁₀-01
  U[DPT]=-4.998₁₀-01
```

```
-4.923967₁₀+00    -9.995₁₀-01     4.924₁₀+00
-4.940469₁₀+00    -9.997₁₀-01     4.940₁₀+00
-4.952979₁₀+00    -9.998₁₀-01     4.953₁₀+00
-4.962531₁₀+00    -9.999₁₀-01     4.963₁₀+00
-4.969877₁₀+00    -9.999₁₀-01     4.970₁₀+00
-4.975571₁₀+00    -1.000₁₀+00     4.976₁₀+00



-4.980017₁₀+00    -1.000₁₀+00     4.980₁₀+00
-4.983518₁₀+00    -1.000₁₀+00     4.984₁₀+00
-4.986296₁₀+00    -1.000₁₀+00     4.986₁₀+00
-4.988518₁₀+00    -1.000₁₀+00     4.989₁₀+00
-4.990309₁₀+00    -1.000₁₀+00     4.990₁₀+00
-4.991764₁₀+00    -1.000₁₀+00     4.992₁₀+00
-4.992956₁₀+00    -1.000₁₀+00     4.993₁₀+00
-4.993938₁₀+00    -1.000₁₀+00     4.994₁₀+00
-4.994753₁₀+00    -1.000₁₀+00     4.995₁₀+00
-4.995435₁₀+00    -1.000₁₀+00     4.995₁₀+00
-4.996008₁₀+00    -1.000₁₀+00     4.996₁₀+00
-4.996493₁₀+00    -1.000₁₀+00     4.996₁₀+00
-4.996906₁₀+00    -1.000₁₀+00     4.997₁₀+00
-4.997259₁₀+00    -1.000₁₀+00     4.997₁₀+00
-4.997563₁₀+00    -1.000₁₀+00     4.998₁₀+00
-4.997825₁₀+00    -1.000₁₀+00     4.998₁₀+00
```

```
      94     1.760₁₀-10       9.999999₁₀-01
  U[DPT]=-4.999₁₀-01
      95     1.805₁₀-10       9.999999₁₀-01
  U[DPT]=-4.999₁₀-01
      96     1.850₁₀-10       9.999999₁₀-01
  U[DPT]=-4.999₁₀-01
      97     1.895₁₀-10       9.999999₁₀-01
  U[DPT]=-4.999₁₀-01
      98     1.940₁₀-10       9.999999₁₀-01
  U[DPT]=-4.999₁₀-01
      99     1.985₁₀-10       9.999999₁₀-01
  U[DPT]=-4.999₁₀-01
     100     2.030₁₀-10       9.999999₁₀-01
  U[DPT]=-4.999₁₀-01
     101     2.075₁₀-10       9.999999₁₀-01
  U[DPT]=-4.999₁₀-01
     102     2.120₁₀-10       9.999999₁₀-01
  U[DPT]=-4.999₁₀-01
     103     2.165₁₀-10       9.999999₁₀-01
  U[DPT]=-4.999₁₀-01
  T=-1.000₁₀+00  NPTS=  104
  NO MORE TRANSIENTS AND STEADY STATE
```

```
 -4.998052₁₀+00   -1.000₁₀+00    4.998₁₀+00

 -4.998251₁₀+00   -1.000₁₀+00    4.998₁₀+00

 -4.998424₁₀+00   -1.000₁₀+00    4.998₁₀+00

 -4.998576₁₀+00   -1.000₁₀+00    4.999₁₀+00

 -4.998711₁₀+00   -1.000₁₀+00    4.999₁₀+00

 -4.998829₁₀+00   -1.000₁₀+00    4.999₁₀+00

 -4.998934₁₀+00   -1.000₁₀+00    4.999₁₀+00

 -4.999028₁₀+00   -1.000₁₀+00    4.999₁₀+00

 -4.999111₁₀+00   -1.000₁₀+00    4.999₁₀+00

 -4.999186₁₀+00   -1.000₁₀+00    4.999₁₀+00


REACHED
```

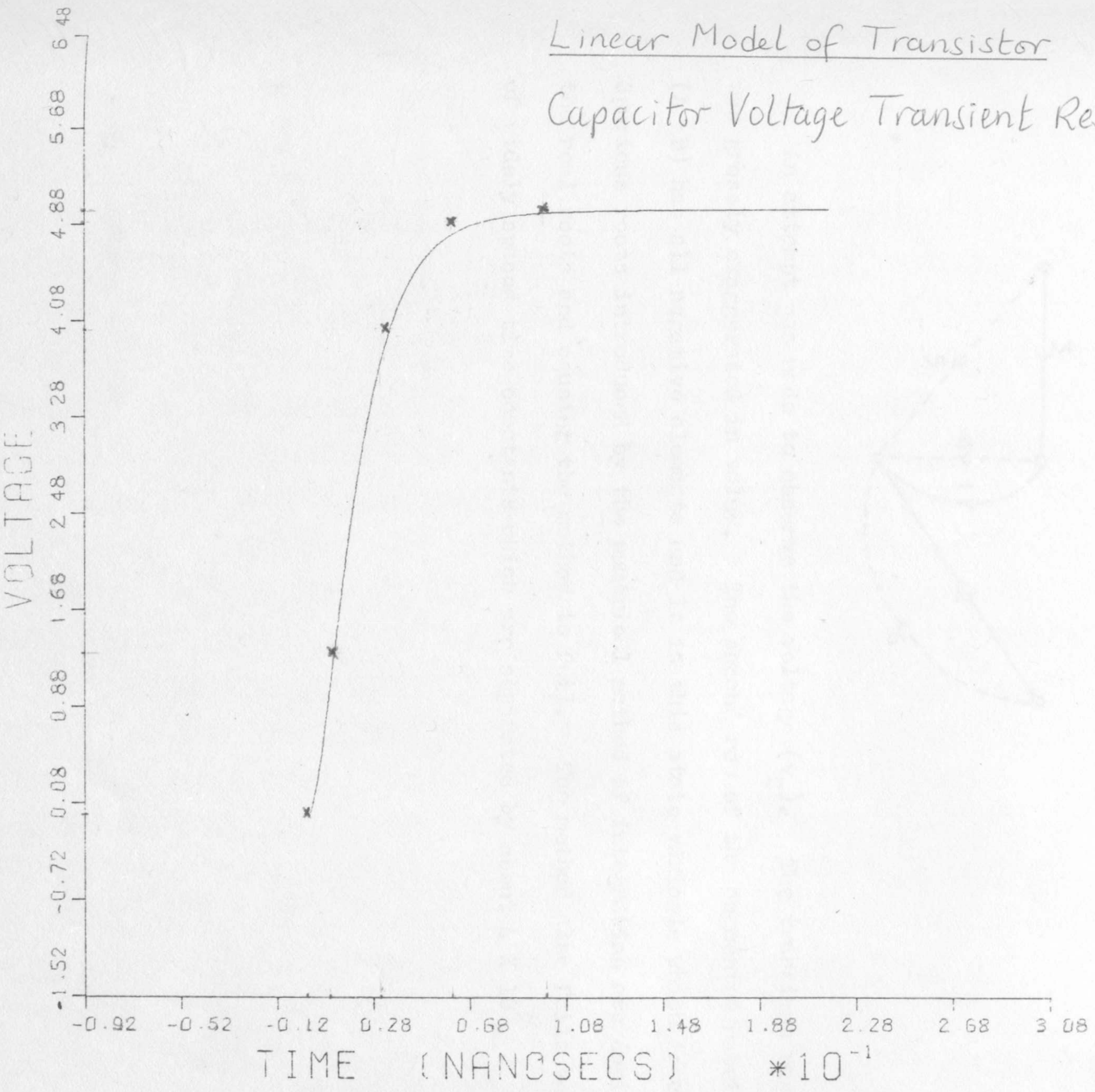


Linear Model of Transistor

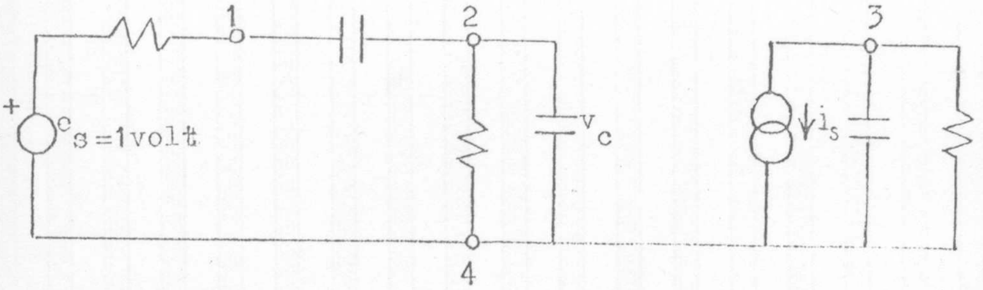
Capacitor Voltage Transient Response (V1)
linearly controlling Current Source (see p. 72)

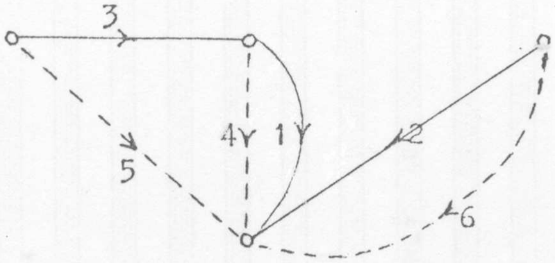
Linear Model of Transistor

Capacitor Voltage Transient Response (V2)



Ex. 13. Non-linear Model of Transistor





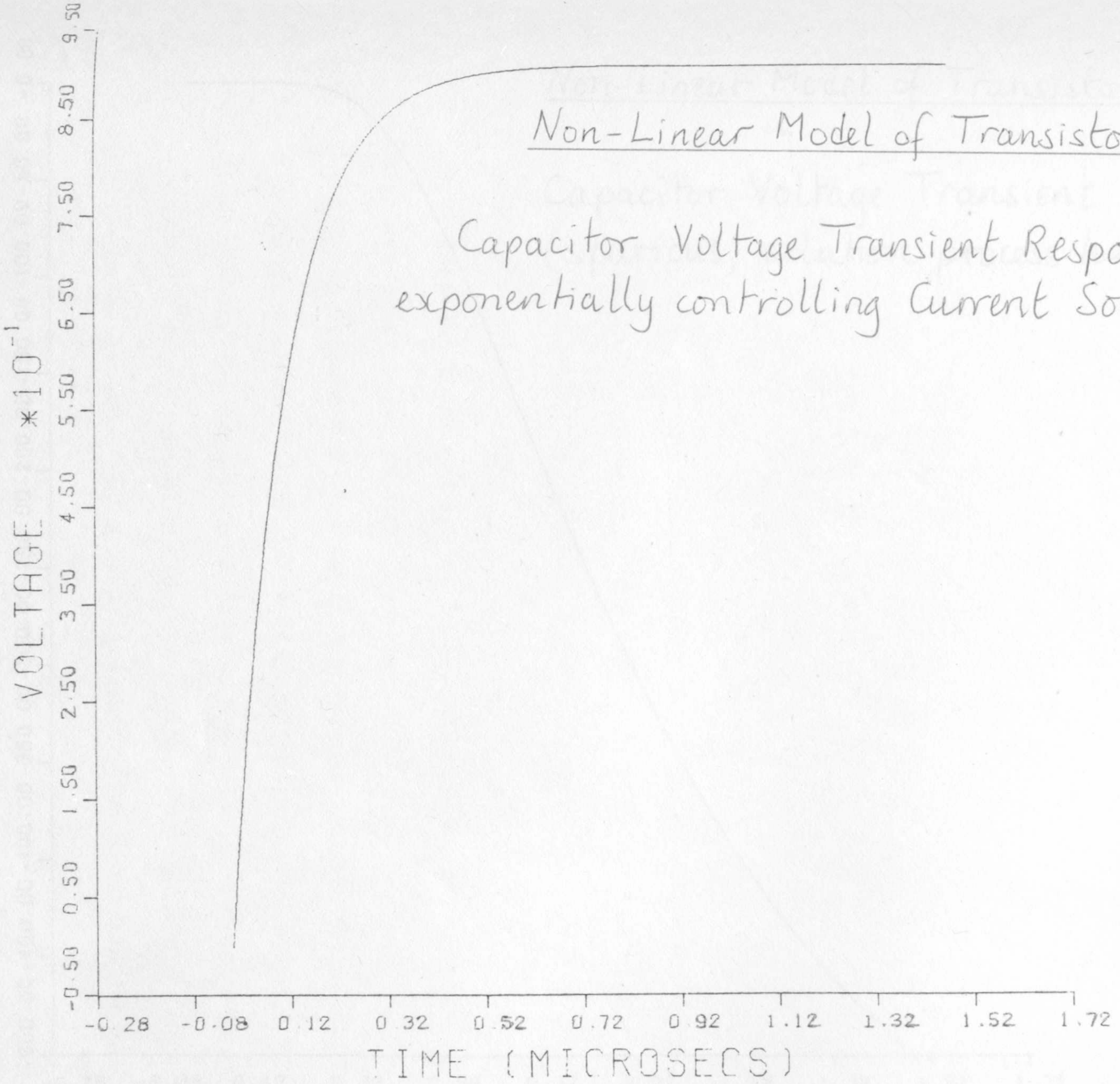
An attempt was made to observe the voltage (v_2). The transient response was grossly exaggerated in value. The second row of the augmented matrix (A B) has all negative elements and it is this state variable which is unstable Spurious roots introduced by the numerical method of integration are dominating the real roots and causing the method to fail. The method also fails because of widely spaced time constants which are separated by about 4 X 10^2.

```
NON-LINEAR MODEL  OF  TRANSISTOR
FOFSVM5 OUTPUT
NRUN=    7

INPUT FOFSVM5;

B=     6   RSOURCE=     2   NY=     2
   B   MN1[B]  MN2[B]  MCOMP[B] ACOMP[B]

   1      2      4      1     2.500⏨-10
   2      3      4      1     1.000⏨-10
   3      4      2      2     5.000⏨+03
   4      1      4      2     5.000⏨+02
   5      2      1      1     1.000⏨-07
   6      4      3      2     5.000⏨+03
     SMN1    SMN2  SMCOMP    SCOL   MSOURCE
        3       4       0       6        2
        4       1       1       4        0

N=    3   L=    3   R=    3   C=    3

OUTPUT FOFSVM5;
ORDER=    3   LC=    0   LR=    0   LL=    0   NC=    0   NR=    0   NL=    0

BETAF[L,N]=
   -1      0      0
   -1      0     -1
    0     -1      0

OUTPUT  FOFSVM5;

TB1[N]=       1        2        5

TL1[L]=       3        4        6
NETWORK  BRANCHES  REORDERED  INTO  TREE  BRANCHES  AND  LINKS

        B   MN1[B] MN2[B] MCOMP[B] ACOMP[B]
        1      2      4      1     2.500⏨-10    V[    1]
        2      3      4      2     1.000⏨-10    V[    2]
        3      1      2      3     1.000⏨-07    V[    3]
        4      2      4     51     5.000⏨+03    J[    4]
        5      1      4     52     5.000⏨+02    J[    5]
        6      3      4     53     5.000⏨+03    J[    6]

     SMN1  SMN2  SMCOMP  SCOL  MSOURCE
```

```
           3          4          0          6          2
           4          1          1          5          0

    SMALLEST OVER LARGEST PIVOT= 1.000₁₀-01
    THE  STATE  VARIABLES  ARE
    (   V[   1]   V[   2]   V[   3])

    A=
       -8.800₁₀+06     0.000₁₀+00    -8.000₁₀+06

        0.000₁₀+00    -2.000₁₀+06     0.000₁₀+00

       -2.000₁₀+04     0.000₁₀+00    -2.000₁₀+04

    B=
        0.000₁₀+00     8.000₁₀+06

       -1.000₁₀+10     0.000₁₀+00

        0.000₁₀+00     2.000₁₀+04

    C=
        1.000₁₀+00     0.000₁₀+00     0.000₁₀+00

        0.000₁₀+00     1.000₁₀+00     0.000₁₀+00

    D=
        0.000₁₀+00     0.000₁₀+00

        0.000₁₀+00     0.000₁₀+00


    WHERE  DX/DT=AX+BU    AND  Y=CX+DU

    H= 5.000₁₀-09
    NINT=       1
    IS= 5.000₁₀-11  THETA= 2.641₁₀+01
    TINT= 0.000₁₀+00   UINT= 1.000₁₀+00
    TINT= 1.000₁₀+06   UINT= 1.000₁₀+00
    EPS= 1.000₁₀-06  EPS1= 2.500₁₀-02  EPS2= 1.000₁₀-01
    T0= 0.000₁₀+00   TN= 4.500₁₀-06   NPTS=        300

      F=0  T= 0.000₁₀+00     0.000₁₀+00     0.000₁₀+00     0.000₁₀+00     0.000₁₀+00     0.000₁₀+00
    VMAX= 1.600₁₀+00   IMAX= 2.000₁₀-03
```

```
STEP SIZE INCREASED    H= 2.500₁₀-08

U[EXP]= 1.127₁₀+00
   280      9.230₁₀-07      9.056198₁₀-01     -3.073144₁₀+03
U[EXP]= 1.136₁₀+00
   281      9.480₁₀-07      9.056309₁₀-01     -3.200189₁₀+03
U[EXP]= 1.143₁₀+00
   282      9.730₁₀-07      9.056317₁₀-01     -3.322939₁₀+03
U[EXP]= 1.150₁₀+00
   283      9.980₁₀-07      9.056242₁₀-01     -3.441377₁₀+03
U[EXP]= 1.156₁₀+00
   284      1.023₁₀-06      9.056101₁₀-01     -3.555512₁₀+03
U[EXP]= 1.162₁₀+00
   285      1.048₁₀-06      9.055906₁₀-01     -3.665377₁₀+03
U[EXP]= 1.166₁₀+00
   286      1.073₁₀-06      9.055669₁₀-01     -3.771023₁₀+03
U[EXP]= 1.170₁₀+00
   287      1.098₁₀-06      9.055397₁₀-01     -3.872515₁₀+03
U[EXP]= 1.174₁₀+00
   288      1.123₁₀-06      9.055098₁₀-01     -3.969931₁₀+03
U[EXP]= 1.177₁₀+00
   289      1.148₁₀-06      9.054777₁₀-01     -4.063360₁₀+03
U[EXP]= 1.180₁₀+00
   290      1.173₁₀-06      9.054438₁₀-01     -4.152897₁₀+03
U[EXP]= 1.182₁₀+00
   291      1.198₁₀-06      9.054085₁₀-01     -4.238645₁₀+03
U[EXP]= 1.184₁₀+00
   292      1.223₁₀-06      9.053720₁₀-01     -4.320710₁₀+03
U[EXP]= 1.186₁₀+00
   293      1.248₁₀-06      9.053347₁₀-01     -4.399201₁₀+03
U[EXP]= 1.188₁₀+00
   294      1.273₁₀-06      9.052966₁₀-01     -4.474232₁₀+03
U[EXP]= 1.189₁₀+00
   295      1.298₁₀-06      9.052579₁₀-01     -4.545914₁₀+03
U[EXP]= 1.190₁₀+00
   296      1.323₁₀-06      9.052187₁₀-01     -4.614361₁₀+03
U[EXP]= 1.191₁₀+00
   297      1.348₁₀-06      9.051792₁₀-01     -4.679686₁₀+03
U[EXP]= 1.191₁₀+00
   298      1.373₁₀-06      9.051394₁₀-01     -4.741999₁₀+03
STEP SIZE INCREASED    H= 5.000₁₀-08

U[EXP]= 1.193₁₀+00
   299      1.423₁₀-06      9.050593₁₀-01     -4.858173₁₀+03
U[EXP]= 1.193₁₀+00
   300      1.473₁₀-06      9.049785₁₀-01     -4.963572₁₀+03
T=-1.000₁₀+00  NPTS=  301
```

```
3.731971⏨-03     9.056⏨-01     -3.073⏨+03

3.777280⏨-03     9.056⏨-01     -3.200⏨+03

3.822563⏨-03     9.056⏨-01     -3.323⏨+03

3.867826⏨-03     9.056⏨-01     -3.441⏨+03

3.913072⏨-03     9.056⏨-01     -3.556⏨+03

3.958303⏨-03     9.056⏨-01     -3.665⏨+03

4.003523⏨-03     9.056⏨-01     -3.771⏨+03

4.048733⏨-03     9.055⏨-01     -3.873⏨+03

4.093934⏨-03     9.055⏨-01     -3.970⏨+03

4.139129⏨-03     9.055⏨-01     -4.063⏨+03

4.184317⏨-03     9.054⏨-01     -4.153⏨+03

4.229500⏨-03     9.054⏨-01     -4.239⏨+03

4.274678⏨-03     9.054⏨-01     -4.321⏨+03

4.319853⏨-03     9.053⏨-01     -4.399⏨+03

4.365023⏨-03     9.053⏨-01     -4.474⏨+03

4.410190⏨-03     9.053⏨-01     -4.546⏨+03

4.455355⏨-03     9.052⏨-01     -4.614⏨+03

4.500516⏨-03     9.052⏨-01     -4.680⏨+03

4.545674⏨-03     9.051⏨-01     -4.742⏨+03



4.635984⏨-03     9.051⏨-01     -4.858⏨+03

4.726284⏨-03     9.050⏨-01     -4.964⏨+03
```

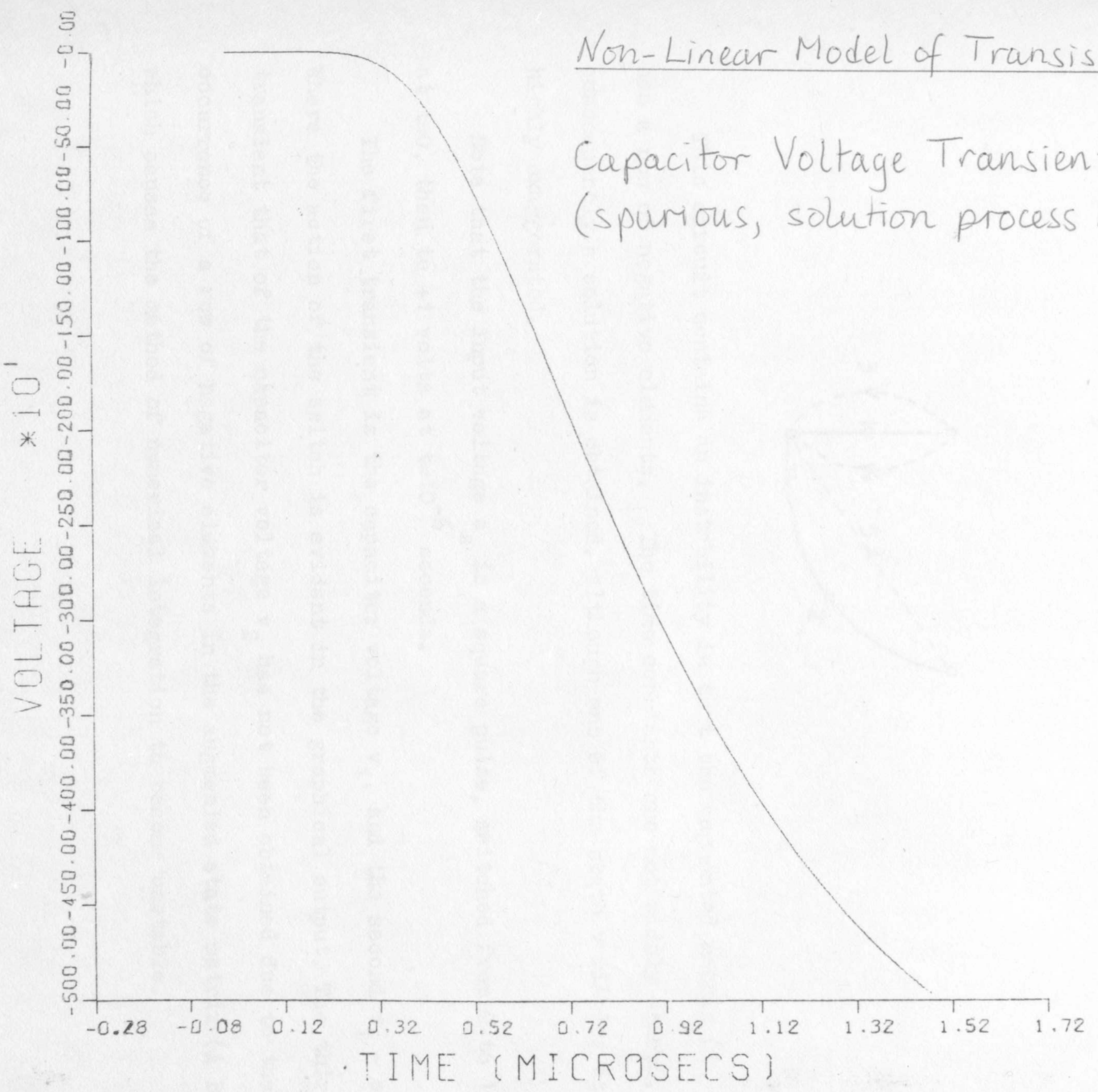


Non-Linear Model of Transistor

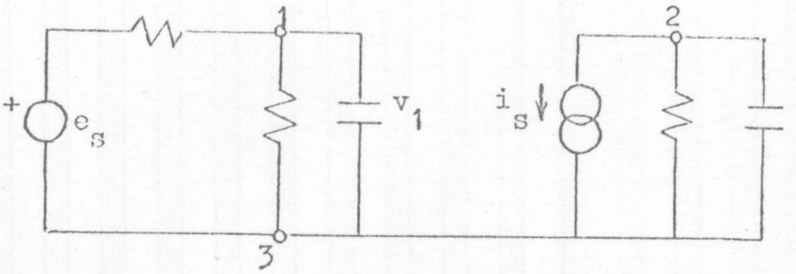
Capacitor Voltage Transient Response (V1) exponentially controlling Current Source (see p: 73)

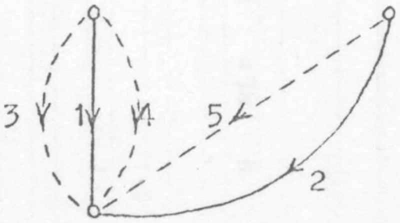
Non-Linear Model of Transistor

Capacitor Voltage Transient Response (V2)
(spurious, solution process having failed)



Ex. 14. Exponential Model of Transistor





This circuit contains an instability in that the augmented matrix (A B) has a row of negative elements. The time constants are not widely spaced, consequently a solution is obtained, although one of the state variables is highly exaggerated.

Note that the input voltage e_s is a square pulse, switched from 0 to 1 volt at t=0, then to +1 volts at $t=10^{-6}$ seconds.

The first transient is the capacitor voltage v_1, and the second $v_1 - e_s$. Where the action of the switch is evident in the graphical output. The third transient that of the capacitor voltage v_2 has not been obtained due to the occurrence of a row of negative elements in the augmented state matrix (A B) which causes the method of numerical integration to become unstable.

```
EXPONENTIAL MODEL OF TRANSISTOR
FOFSVM5 OUTPUT
NRUN=  -1

INPUT FOFSVM5;

B=    5   RSOURCE=     2   NY=     3
    B    MN1[B]  MN2[B]  MCOMP[B] ACOMP[B]

    1      1       3        1    5.000₁₀-10
    2      1       3        2    1.000₁₀+03
    3      2       3        1    1.000₁₀-10
    4      1       3        2    1.000₁₀+03
    5      2       3        2    1.000₁₀+03
      SMN1    SMN2  SMCOMP    SCOL  MSOURCE
         2       3       0       5       2
         3       1       1       2       0

N=    2   L=    3   R=    3   C=    2


OUTPUT FOFSVM5;
ORDER=    2   LC=    0   LR=    1   LL=    0   NC=    0   NR=    0   NL=    0

RETAF[L,N]=
    -1     0
    -1     0
     0    -1

OUTPUT  FOFSVM5;


TB1[N]=        1        3

TL1[L]=        2        4        5
NETWORK  BRANCHES  REORDERED  INTO  TREE  BRANCHES  AND  LINKS


        B   MN1[B] MN2[B] MCOMP[B] ACOMP[B]
        1      1      3      1    5.000₁₀-10    V[    1]
        2      2      3      2    1.000₁₀-10    V[    2]
        3      1      3     51    1.000₁₀+03    J[    3]
        4      1      3     52    1.000₁₀+03    J[    4]
        5      2      3     53    1.000₁₀+03    J[    5]

      SMN1  SMN2  SMCOMP  SCOL  MSOURCE
         2     3       0     5       2
         3     1       1     3       0
```

```
TB3=
      4       5
TL3=
      1       2       3

BETAF[L,N]=
    -1     0
     0    -1
    -1     0
SMALLEST OVER LARGEST PIVOT= 1.000₁₀+00
SMALLEST OVER LARGEST PIVOT= 1.000₁₀+00
THE  STATE  VARIABLES  ARE
(   V[   1]   V[   2])

A=
    -4.000₁₀+06     0.000₁₀+00

     0.000₁₀+00    -1.000₁₀+07

B=
     0.000₁₀+00     2.000₁₀+06

    -1.000₁₀+10     0.000₁₀+00

C=
     1.000₁₀+00     0.000₁₀+00

     1.000₁₀+00     0.000₁₀+00

     0.000₁₀+00     1.000₁₀+00

D=
     0.000₁₀+00     0.000₁₀+00

     0.000₁₀+00    -1.000₁₀+00

     0.000₁₀+00     0.000₁₀+00


WHERE  DX/DT=AX+BU   AND  Y=CX+DU

H= 1.000₁₀-08
NINT=        2
```

```
IS= 1.000₁₀-09  THETA= 3.961₁₀+01
TINT= 0.000₁₀+00   UINT=-1.000₁₀+00
TINT= 1.000₁₀-06   UINT=-1.000₁₀+00
TINT= 1.000₁₀-06   UINT= 1.000₁₀+00
TINT= 1.000₁₀+30   UINT= 1.000₁₀+00
EPS= 1.000₁₀-06  EPS1= 2.500₁₀-02  EPS2= 1.000₁₀-01
T0= 0.000₁₀+00  TN= 1.000₁₀+06  NPTS=       500

  F=0  1= 0.000₁₀+00     0.000₁₀+00     0.000₁₀+00     0.
VMAX= 1.000₁₀+00    IMAX= 1.000₁₀-03

U[EXP]= 0.000₁₀+00
    1     1.000₁₀-08   -1.960528₁₀-02    0.000000₁₀+00
U[EXP]=-4.669₁₀-10
    2     2.000₁₀-08   -3.844183₁₀-02    4.443383₁₀-08
U[EXP]=-6.988₁₀-10
    3     3.000₁₀-08   -5.653978₁₀-02    1.067096₁₀-07
U[EXP]=-8.219₁₀-10
    4     4.000₁₀-08   -7.392811₁₀-02    1.747673₁₀-07
U[EXP]=-8.905₁₀-10
    5     5.000₁₀-08   -9.063462₁₀-02    2.428793₁₀-07
U[EXP]=-9 304₁₀-10
    6     6.000₁₀-08   -1.066861₁₀-01    3.083014₁₀-07
U[EXP]=-9.543₁₀-10
    7     7.000₁₀-08   -1.221081₁₀-01    3.697772₁₀-07
U[EXP]=-9.692₁₀-10
    8     8.000₁₀-08   -1.369255₁₀-01    4.268166₁₀-07
U[EXP]=-9.786₁₀-10
    9     9.000₁₀-08   -1.511618₁₀-01    4.793302₁₀-07
U[EXP]=-9.849₁₀-10
   10     1.000₁₀-07   -1.648400₁₀-01    5.274376₁₀-07
STEP SIZE INCREASED    H= 2.000₁₀-08

U[EXP]=-9.915₁₀-10
   11     1.200₁₀-07   -1.906083₁₀-01    6.115606₁₀-07
U[EXP]=-9.950₁₀-10
   12     1.400₁₀-07   -2.143955₁₀-01    6.810627₁₀-07
U[EXP]=-9.969₁₀-10
   13     1.600₁₀-07   -2.363538₁₀-01    7.383096₁₀-07
U[EXP]=-9.980₁₀-10
   14     1.800₁₀-07   -2.566239₁₀-01    7.853770₁₀-07
U[EXP]=-9.986₁₀-10
   15     2.000₁₀-07   -2.753355₁₀-01    8.240319₁₀-07
U[EXP]=-9.990₁₀-10
   16     2.200₁₀-07   -2.926085₁₀-01    8.557550₁₀-07
U[EXP]=-9.993₁₀-10
   17     2.400₁₀-07   -3.085536₁₀-01    8.817766₁₀-07
U[EXP]=-9.995₁₀-10
```

```
000₁₀+00      0.000₁₀+00      0.000₁₀+00


   -1.961₁₀-02      9.804₁₀-01      0.000₁₀+00

   -3.844₁₀-02      9.616₁₀-01      4.443₁₀-08

   -5.654₁₀-02      9.435₁₀-01      1.067₁₀-07

   -7.393₁₀-02      9.261₁₀-01      1.748₁₀-07

   -9.063₁₀-02      9.094₁₀-01      2.429₁₀-07

   -1.067₁₀-01      8.933₁₀-01      3.083₁₀-07

   -1.221₁₀-01      8.779₁₀-01      3.698₁₀-07

   -1.369₁₀-01      8.631₁₀-01      4.268₁₀-07

   -1.512₁₀-01      8.488₁₀-01      4.793₁₀-07

   -1.648₁₀-01      8.352₁₀-01      5.274₁₀-07
                               =

   -1.906₁₀-01      8.094₁₀-01      6.116₁₀-07

   -2.144₁₀-01      7.856₁₀-01      6.811₁₀-07

   -2.364₁₀-01      7.636₁₀-01      7.383₁₀-07

   -2.566₁₀-01      7.434₁₀-01      7.854₁₀-07

   -2.753₁₀-01      7.247₁₀-01      8.240₁₀-07

   -2.926₁₀-01      7.074₁₀-01      8.558₁₀-07

   -3.086₁₀-01      6.914₁₀-01      8.818₁₀-07
```

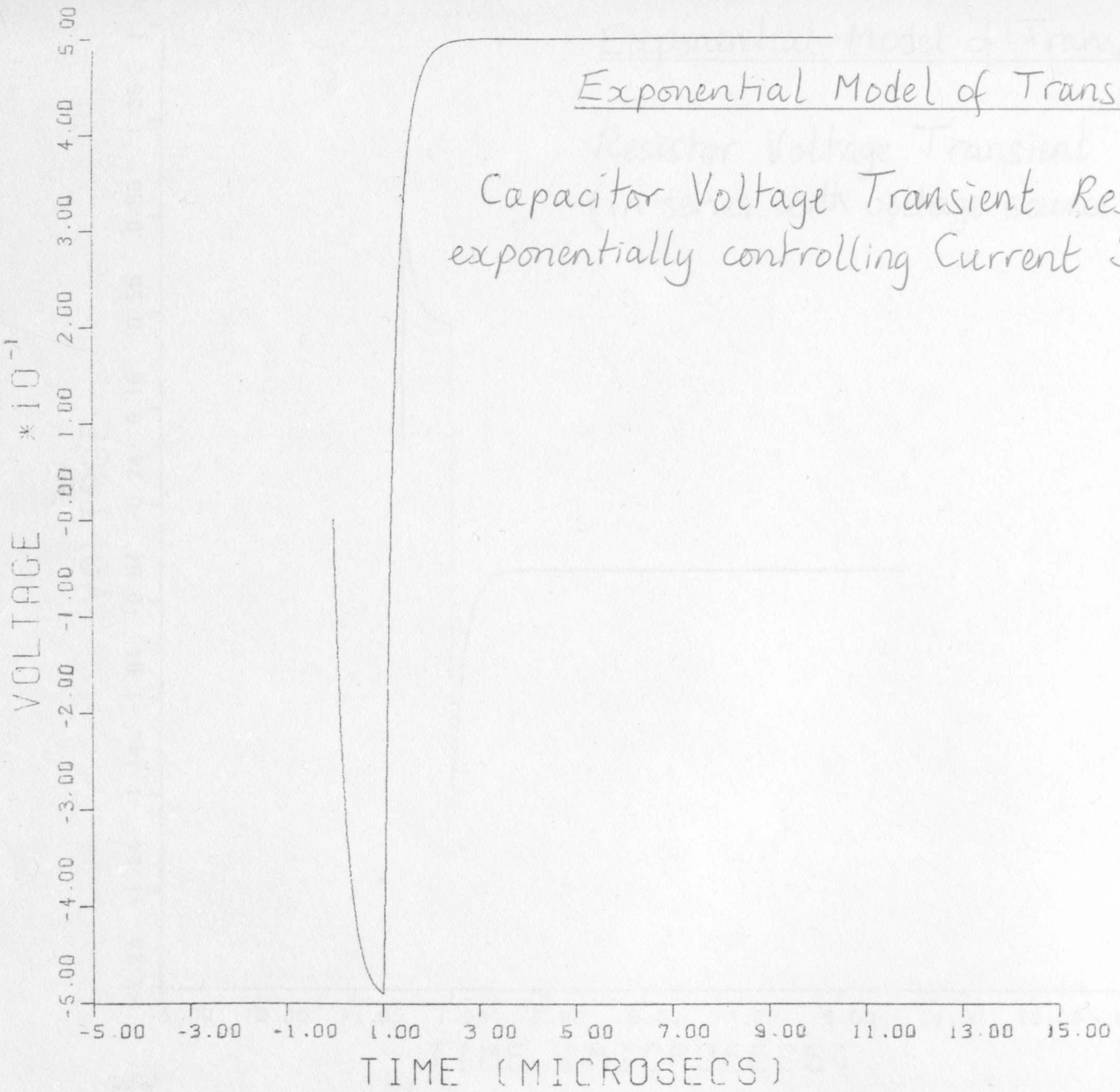
```
    341     9,175₁₀-06     5.000000₁₀-01     -3.976835₁₀+02
 U[EXP]= 3.978₁₀-01
    342     9.225₁₀-06     5.000000₁₀-01     -3.977121₁₀+02
 U[EXP]= 3.978₁₀-01
    343     9.275₁₀-06     5.000000₁₀-01     -3.977400₁₀+02
 U[EXP]= 3.978₁₀-01
    344     9.325₁₀-06     5.000000₁₀-01     -3.977671₁₀+02
 U[EXP]= 3.978₁₀-01
    345     9.375₁₀-06     5.000000₁₀-01     -3.977935₁₀+02
 U[EXP]= 3.979₁₀-01
    346     9.425₁₀-06     5.000000₁₀-01     -3.978191₁₀+02
 U[EXP]= 3.979₁₀-01
    347     9.475₁₀-06     5.000000₁₀-01     -3.978442₁₀+02
 U[EXP]= 3.979₁₀-01
    348     9.525₁₀-06     5.000000₁₀-01     -3.978685₁₀+02
 U[EXP]= 3.979₁₀-01
    349     9.575₁₀-06     5.000000₁₀-01     -3.978922₁₀+02
 U[EXP]= 3.980₁₀-01
    350     9.625₁₀-06     5.000000₁₀-01     -3.979154₁₀+02
 U[EXP]= 3,980₁₀-01
    351     9.675₁₀-06     5.000000₁₀-01     -3.979379₁₀+02
 U[EXP]= 3,980₁₀-01
    352     9.725₁₀-06     5.000000₁₀-01     -3.979598₁₀+02
 U[EXP]= 3.980₁₀-01
    353     9.775₁₀-06     5.000000₁₀-01     -3.979812₁₀+02
 U[EXP]= 3.980₁₀-01
    354     9.825₁₀-06     5.000000₁₀-01     -3.980020₁₀+02
 U[EXP]= 3.981₁₀-01
    355     9.875₁₀-06     5.000000₁₀-01     -3.980223₁₀+02
 U[EXP]= 3.981₁₀-01
    356     9.925₁₀-06     5.000000₁₀-01     -3.980421₁₀+02
 U[EXP]= 3.981₁₀-01
    357     9.975₁₀-06     5.000000₁₀-01     -3.980614₁₀+02
 U[EXP]= 3.981₁₀-01
    358     1,002₁₀-05     5.000000₁₀-01     -3.980802₁₀+02
 U[EXP]= 3.981₁₀-01
    359     1,007₁₀-05     5.000000₁₀-01     -3.980986₁₀+02
 U[EXP]= 3.981₁₀-01
    360     1.012₁₀-05     5.000000₁₀-01     -3.981165₁₀+02
 U[EXP]= 3.982₁₀-01
    361     1.017₁₀-05     5.000000₁₀-01     -3.981339₁₀+02
 U[EXP]= 3.982₁₀-01
    362     1.022₁₀-05     5.000000₁₀-01     -3.981510₁₀+02
 U[EXP]= 3,982₁₀-01
    363     1.027₁₀-05     5.000000₁₀-01     -3.981676₁₀+02
 U[EXP]= 3.982₁₀-01
    364     1.032₁₀-05     5.000000₁₀-01     -3.981838₁₀+02
 U[EXP]= 3.982₁₀-01
```

```
5.000⏨-01    -5.000⏨-01    -3.977⏨+02
5.000⏨-01    -5.000⏨-01    -3.977⏨+02
5.000⏨-01    -5.000⏨-01    -3.977⏨+02
5.000⏨-01    -5.000⏨-01    -3.978⏨+02
5.000⏨-01    -5.000⏨-01    -3.978⏨+02
5.000⏨-01    -5.000⏨-01    -3.978⏨+02
5.000⏨-01    -5.000⏨-01    -3.978⏨+02
5.000⏨-01    -5.000⏨-01    -3.979⏨+02
5.000⏨-01    -5.000⏨-01    -3.979⏨+02
5.000⏨-01    -5.000⏨-01    -3.979⏨+02
5.000⏨-01    -5.000⏨-01    -3.979⏨+02
5.000⏨-01    -5.000⏨-01    -3.980⏨+02
5.000⏨-01    -5.000⏨-01    -3.980⏨+02
5.000⏨-01    -5.000⏨-01    -3.980⏨+02
5.000⏨-01    -5.000⏨-01    -3.980⏨+02
5.000⏨-01    -5.000⏨-01    -3.980⏨+02
5.000⏨-01    -5.000⏨-01    -3.981⏨+02
5.000⏨-01    -5.000⏨-01    -3.981⏨+02
5.000⏨-01    -5.000⏨-01    -3.981⏨+02
5.000⏨-01    -5.000⏨-01    -3.981⏨+02
5.000⏨-01    -5.000⏨-01    -3.981⏨+02
5.000⏨-01    -5.000⏨-01    -3.982⏨+02
5.000⏨-01    -5.000⏨-01    -3.982⏨+02
5.000⏨-01    -5.000⏨-01    -3.982⏨+02
```

```
   365    1.037₁₀-05    5.000000₁₀-01    -3.981997₁₀+02
 U[EXP]= 3.982₁₀-01
 T=-1.000₁₀+00  NPTS=  366
 NO MORE TRANSIENTS AND STEADY STATE REACHED
```

LR

```
5.000⏨-01   -5.000⏨-01   -3.982⏨+02
```

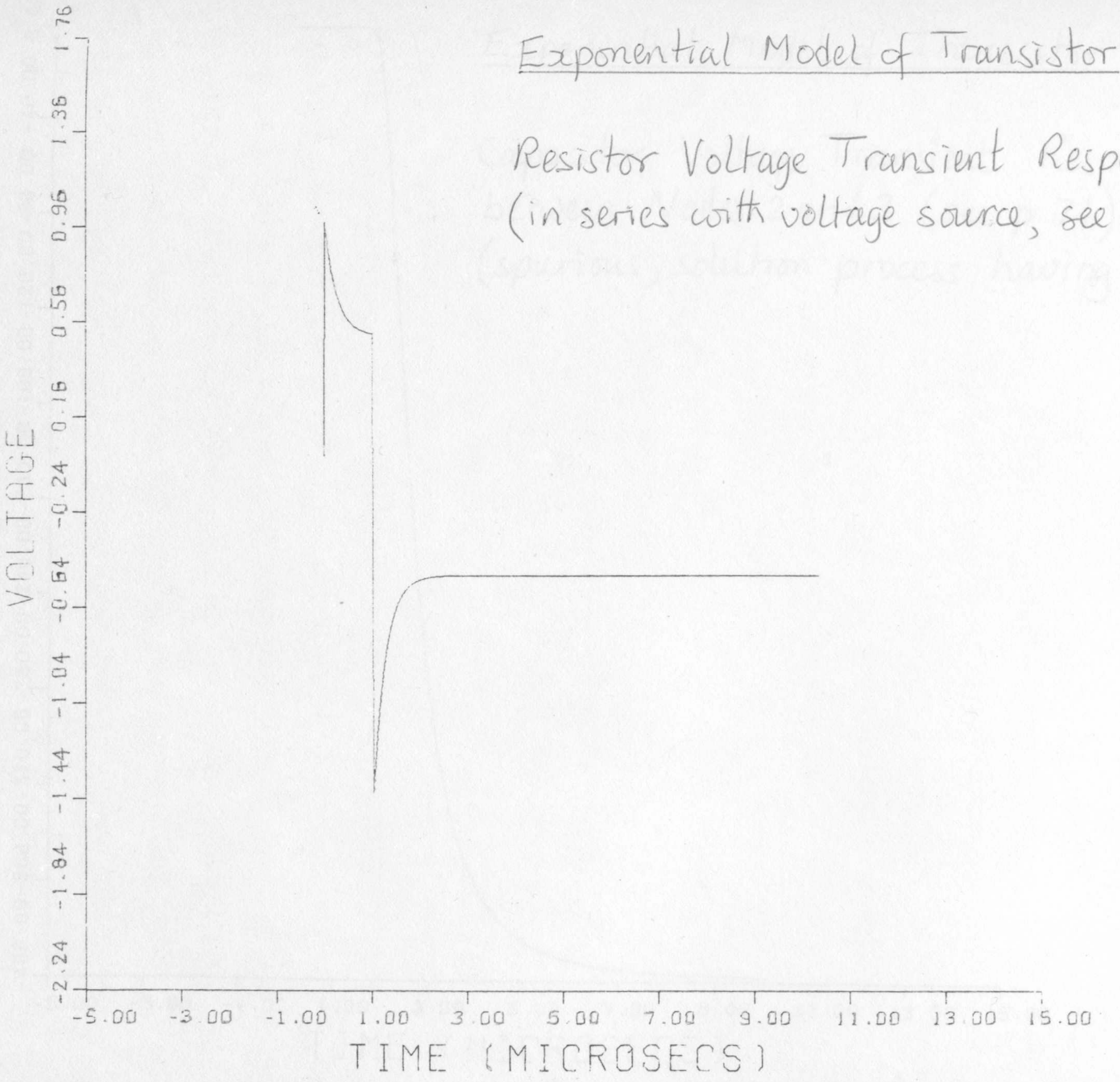


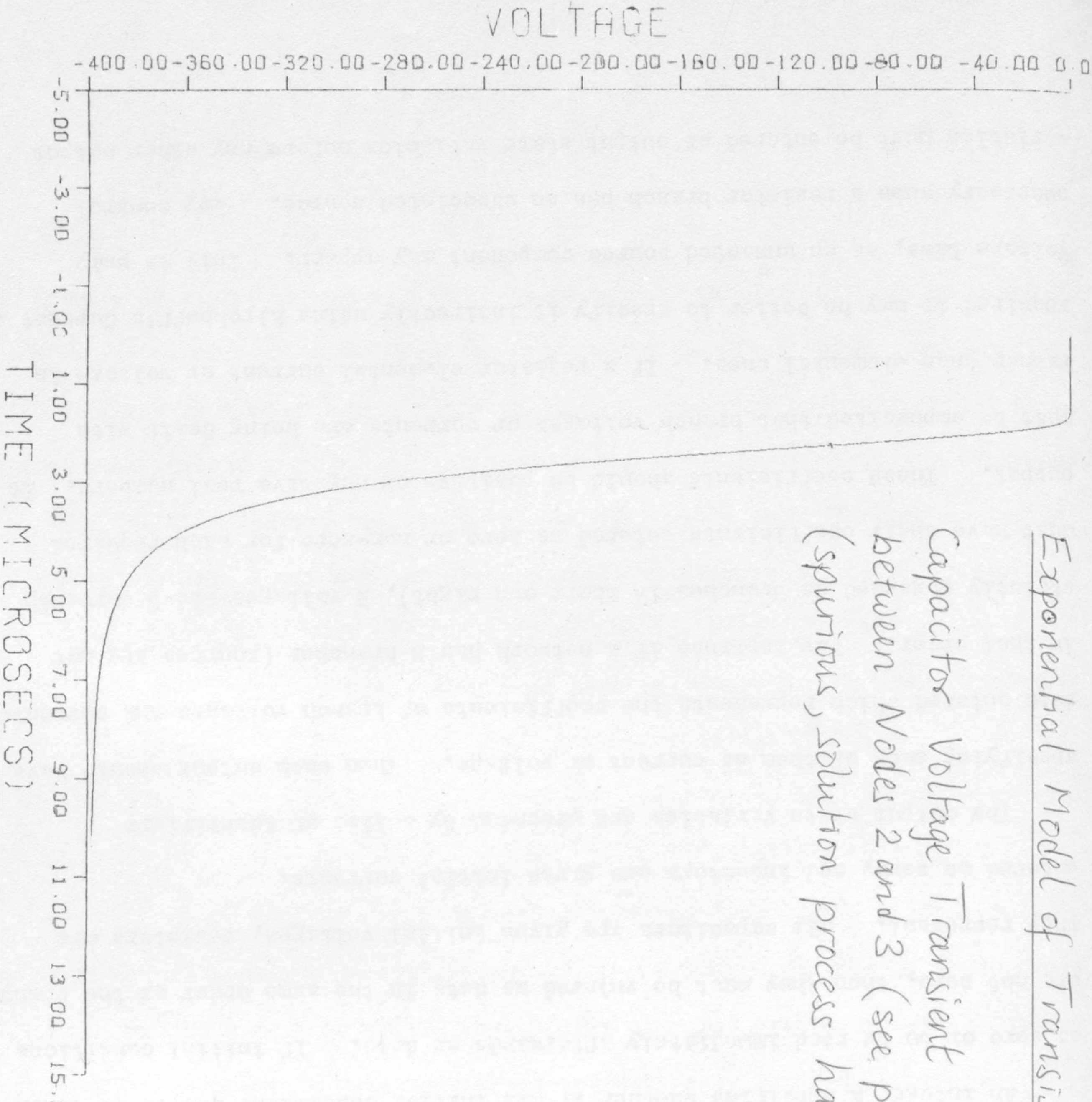
Exponential Model of Transistor

Capacitor Voltage Transient Response (V1)
exponentially controlling Current Source (see p. 74)

Exponential Model of Transistor

Resistor Voltage Transient Response
(in series with voltage source, see p. 74)





Exponential Model of Transistor Capacitor Voltage Transient Response between Nodes 2 and 3 (see p. 74) (spurious, solution process having failed)

5.5 Operating Instructions

Data for use with program FOFSVM5 is arranged as if for a single run, the individual problems being separated from each other by identifying names and a run number which when negative signifies that no more data is to be read.

The branches of a network are identified by numbered nodes which are used as coordinates, and two more numbers which designate the type of element contained in a branch and its value. Source branches are identified similarly by node coordinates and two numbers which state whether it is current or voltage and its type of dependence. Dependent sources should be entered first followed by anyindependent sources.

An integer K specifies whether if all initial conditions are to be taken as zero or to be read immediately afterwards as data. If initial conditions are not zero, then they must be entered as data in the same order as the branches they represent. All capacitors are given initial voltages, resistors are entered as zero, and inductors are given initial currents.

The output state variables are preceded by a list of identifiers specifying each of them as current or voltage. Then each output should have data entered which represents the coefficients of branch voltages and currents in that order. For instance if a network has B branches (sources are not strictly regarded as branches in their own right), B voltages and B currents must have their coefficients entered as zero or non-zero for each required output. These coefficients should be positive or negative real numbers. It must be emphasized that branch voltages or currents are being dealt with rather than elemental ones. If a resistor elemental current or voltage is required it may be better to specify it indirectly using Kirchhoff's Current or Voltage Laws, as an unwanted source component may appear. This is only necessary when a resistor branch has an associated source. Any control variables must be entered as output state variables before any other output requirements. Also these output states representing control variables must be entered in the same order as the sources they pertain to have been entered as data.

Other data related to the sources is also required and must be entered in the same order as the previous source data which was related to the source columns of the state matrix. This latter source data pertains to the input vector u. Again dependent sources are entered first in the same order as they were before, followed by the independent sources. It is probably best to enter the sources all the way through the following order; sinusoidal, exponentially controlled, linearly controlled, and independent.

Independent sources have time intervals specified by the user. The number of time intervals for each independent source must be entered. The coordinates of the end-points of the graphs of independent sources are entered. For instance, a single step function would have

$$((t_{\emptyset},u_{\emptyset}),(t_1,u_{\emptyset}))$$

whereas a single ramp would have

$$((t_{\emptyset},u_{\emptyset}),(t_1,u_1))$$

a ramp followed by a plateau followed by a ramp would have

$$((t_{\emptyset},u_{\emptyset}),(t_1,u_1));\ ((t_2,u_1),(t_3,u_1));\ ((t_4,u_1),(t_5,u_2)).$$

Finally a set of numbers is entered which are the state matrix series truncation factor, local error tolerance limits, time interval over which responses are to be observed, maximum number of points on response curves are to be plotted, and an indicator whether or not a full printout of state variables and output state variables or just output state variables alone are required on the lineprinter. (See section 5.1 for formal layout of data input.)

5.6 Operator Procedure

FOFSVM5 needs two magnetic tapes for use during the running of the program. One is used for the segmentation of the program and the other to dump data for use with the program SVMPLT which can be put into batch with FOFSVM5. The first is an initialised scratch tape mounted on handler 3 and the second is initialised, called 'FOFSVM OUTPUT' and mounted on handler 6.

After FOFSVM5 has completed its runs SVMPLT reads the tape on handler 6 and then writes data to the same tape. The plotter tape for CALCOMP is then required by the program.

6. Final Conclusions and Future Developments

There are three main tasks that a computer program based on state variable analysis has to perform, namely:-

(i) the formulation of a degenerate system of equations (see section 2.2),

(ii) the processing of these until they eventually become the state equations (see section 2.3),

(iii) the solution of the state equations (see section 3).

The first task requires the computation of a topological tree and its associated tieset matrix. This can be done using either matrix or search methods. A degenerate system of equations can then be formulated.

The degenerate variables can be eliminated using Pottle's Method or the method of Auxiliary Trees which is based on Bashkow's work. The latter method has the advantage in that Gaussian Elimination is applied only once to each subsystem of equations of constraint, which are formulated topologically, whereas the former method uses Gaussian Elimination repeatedly on the considerably larger main system of degenerate equations to formulate the equations of constraint, the process being repeated after each degenerate variable has been eliminated until only the state equations remain.

The solution of the state equations can be accomplished using:-

(i) an Adams-Bashforth-Moulton predictor corrector method,[4]

(ii) a Runge-Kutta method such as England's Method,[4]

(iii) Certaine's Method of transition matrices (see section 3.2),

(iv) Methods suitable for use with systems containing equations with widely separated time constants known as Stiff equations, e.g. Gear's Method (see section 3.3).

It was found that the formulation of the degenerate system of equations was relatively easy to program. In Bashkow's Topological formulation the sources are associated with a branch containing a passive element whereas in Pottle's formulation they are treated as independent branches. It was decided to associate all sources with resistor branches as this simplifies the classification of the dynamic constraints as well as suppressing derivatives and integrals of sources, which could still occur in certain cases, despite the choice of capacitor voltages and inductor currents as state variables. However, associating sources with resistor branches would not prevent derivatives and integrals of the sources appearing if this choice of state variables was not made.

The section dealing with the elimination of the degenerate variables from the main system of equations proved more difficult to program. Programming errors were made which proved difficult to locate and eliminate. Test problems had to be sufficiently large to test the program but small enough to check by hand. Formulations were tried for networks containing from two to fifteen branches. One tree was computed for a network containing thirty branches. The number of branches in a network gives the initial order of the system matrix that has to be manipulated. The method of Auxiliary Trees was chosen to eliminate the constraints.

The solution of the state equations gave rise to problems which were sometimes difficult to diagnose. Some equations could not be solved without a variable step size routine, as too many points were required over the prescribed time interval using the initial step size estimate computed by the program. There was in fact no suitable single step size estimate over the whole range of interest. Other failures originally thought to be due to programming inconsistencies were corrected by monitoring the local numerical errors; this resulted in the solution process being speeded up in most cases. Networks whose components had widely separated time constants caused the solution process to slow down to an unacceptable level, and no way round this problem could be found using Certaine's Method. But apart from this last problem, all test networks containing capacitors, resistors, inductors and independent sources were solved. However test circuits containing linearly controlled sources failed without exception when a row of the augmented matrix (A B) contained all negative elements. This gave rise to transient responses of the right shape but grossly distorted in magnitude. This apparently occurs when a step length cannot be found which lies within the interval of relative stability for the method of numerical integration being used. When an exponentially controlled source was introduced the transient response did not stabilize. It appears that small local numerical errors were being amplified exponentially. The introduction of a damping factor[1] apparently brought this error propagation under control as the transient response then settled down to the right value after some numerical oscillation.

Program FOFSVM5 was based on Bashkow's topological formulation of the state equations and uses the Auxiliary Tree method of eliminating the degenerate variables from the main system of equations. Procedures TREE, TIESET MATRIX, SUBSYSTEM, GFPIVOT and REDUCE MATRIX were used to formulate and process the degenerate system of equations until they become the state equations. TREE and TIESET MATRIX use search methods to compute the trees and their associated tieset matrices. SUBSYSTEM uses these trees and tieset matrices to formulate the subsystems of equations of constraint. GFPIVOT uses Gaussian Elimination with full pivoting followed by back substitution to solve these interactive subsystems of equations one by one along with REDUCE MATRIX which substitutes for the dependent variables in the main system and other subsystem which have yet to be solved. The resistor variables are dealt with simultaneously using the same procedures. The state equations are solved using Certaine's method of numerical integration with variable step size and local error checking routines. The program terminates a run if any of the following conditions occur:-

(i) the solution process is moving too slowly and cannot be speeded up without intolerable local error accumulation,

(ii) a steady-state response has been produced and there are no more excitations to be applied to the network,

(iii) a prescribed maximum number of points to be plotted has been attained,

(iv) the prescribed time interval over which the network is being investigated has been exceeded.

The output state-variable data is put onto magnetic tape and becomes the input for a plotting program SVMPLT which eventually produces graphs of the required transient responses on an off-line device. The state equations can be formulated for general networks containing capacitors, resistors, inductors, independent sources, which may be constant or piecewise-continuous-linear functions of time, dependent sources which may be linearly or exponentially controlled, and sources which vary sinusoidally with time. The state equations can be solved if no widely separated time constants are present and if a step size can be chosen within the interval of relative stability for the method of numerical integration being used.

FOFSVM6 uses matrix methods to simultaneously compute the trees and their associated cutset matrices using a modified form of procedure GFPIVOT, which takes over the tasks previously executed by TREE and TIESET MATRIX. Otherwise the program uses the same techniques as FOFSVM5, but has not been brought to the same stage of development.

It is hoped that in the future program FOFSVM5 can be used to investigate integrated circuits. Any active components are to be replaced by circuit models containing capacitors, resistors, inductors and controlled sources. This active circuit modelling has been left to the user to make the program more versatile in that it can be made more applicable to the kind of circuit behaviour under analysis. The program is already of considerable length and has been segmented to create more run-space. The third segment, which at present implements Certaine's Method[2] of solving the state equations needs to be replaced by a powerful method of solution which will be able to deal with Stiff equations when they occur and which has a larger interval of relative stability. Gear's Method should replace the present method of solution. An extension of the program using charge or flux controlled models, could formulate the state equations of time-varying networks, but solution methods for specific cases only could be produced.

References

1. Zobrist, G.W. 'Network computer analysis.' Boston Technical Publications.

2. Calahan, D.A. 'Computer aided network design.' Rev. Ed. 1972 McGraw-Hill

3. Gear, C.W. 'The automatic integration of ordinary differential equations.' Comm. ACM., 14, 176-179 (1971).

4. Lambert, J.D. 'Computational methods in ordinary differential equations.' Wiley, New York 1973.

5. Murdoch, J.B. 'Network theory.' McGraw-Hill.

6. Gear, C.W. 'The numerical integration of ordinary differential equations.' Math. Comp. 21, 2 (April 1967) 146-156.

7. Gear, C.W. 'The automatic integration of stiff ordinary differential equations.' Information Processing 68, A.J.H. Morrell, Ed., North Holland, Amsterdam, 1969, 187-193.

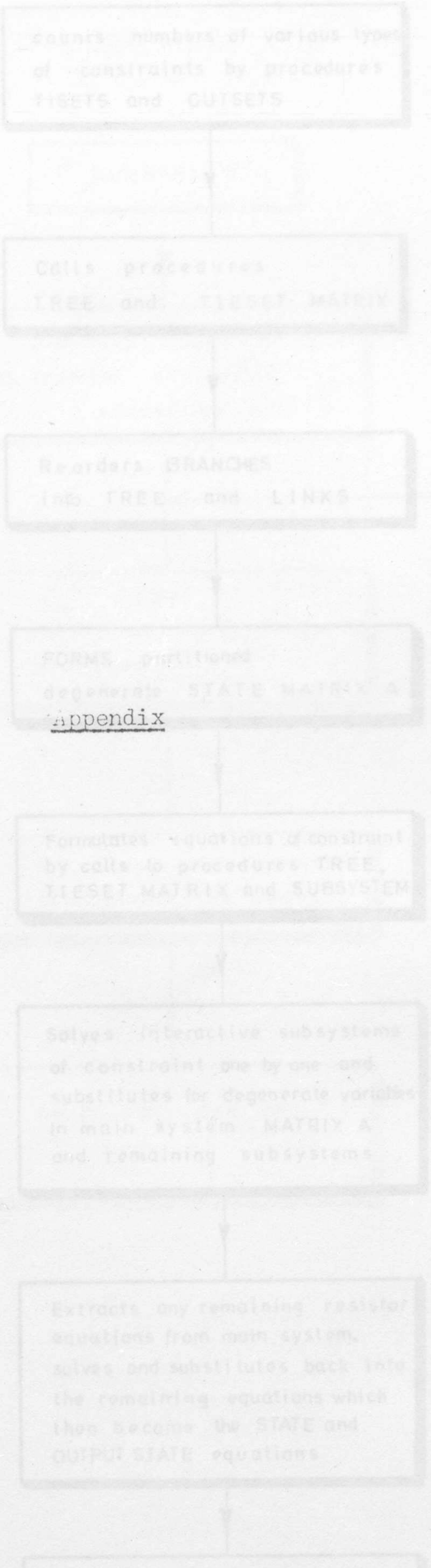
8. Henrici, P. 'Discrete variable methods in ordinary differential equations Wiley, New York, 1962, Ch. 5.

9. Haykin, S.S. 'Active network theory.' Addison-Wesley 1970.

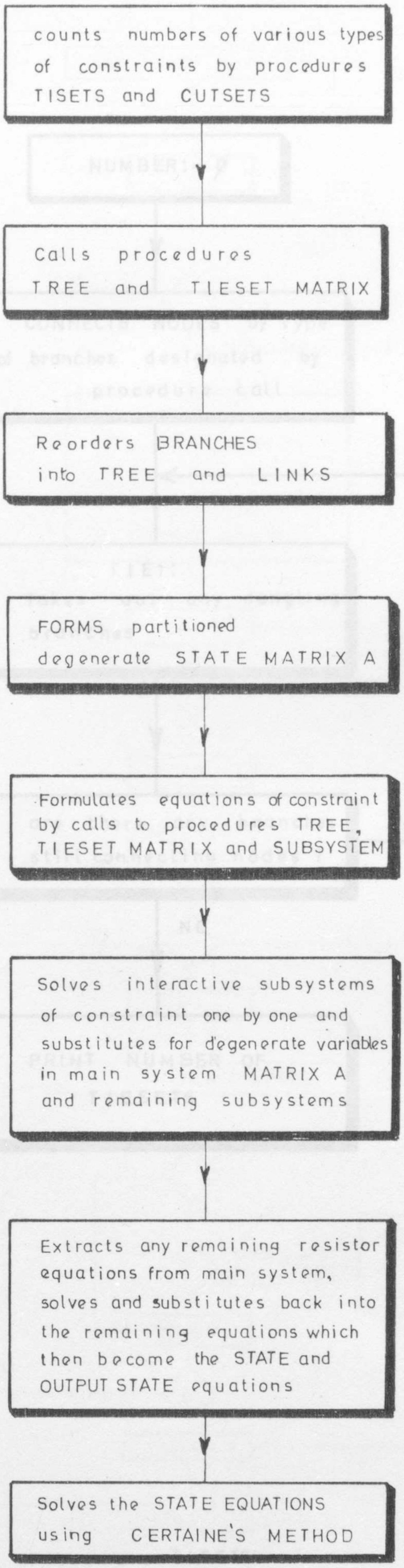
10. Bradshaw, Dr. M.W. 'NESTLE: A General Purpose Nonlinear Network, Transient and D.C. Analysis Program.' Elliott Automation Ltd.

11. Kuo, F.F. and Kaiser, J.K. 'System Analysis by Digital Computer', Wiley New York, 1966, p.112.

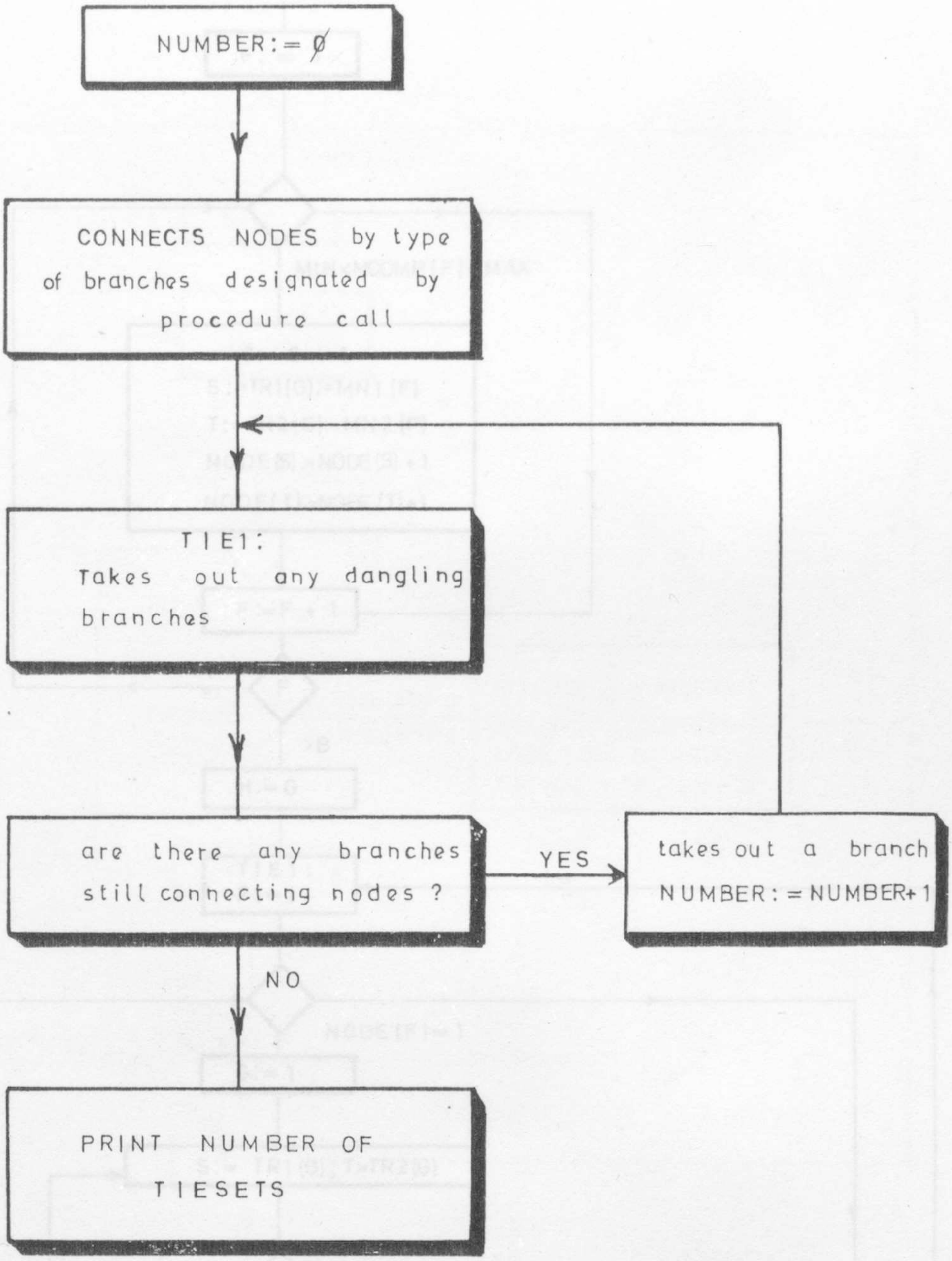
Appendix


counts numbers of various types of constraints by procedures TISETS and CUTSETS
Calls procedures TREE and TIESET MATRIX
Reorders BRANCHES into TREE and LINKS
FORMS partitioned degenerate STATE MATRIX A
Formulates equations of constraint by calls to procedures TREE, TIESET MATRIX and SUBSYSTEM
Solves interactive subsystems of constraint one by one and substitutes for degenerate variables in main system MATRIX A and remaining subsystems
Extracts any remaining resistor equations from main system, solves and substitutes back into the remaining equations which then become the STATE and OUTPUT STATE equations
Solves the STATE EQUATIONS using CERTAINE'S METHOD

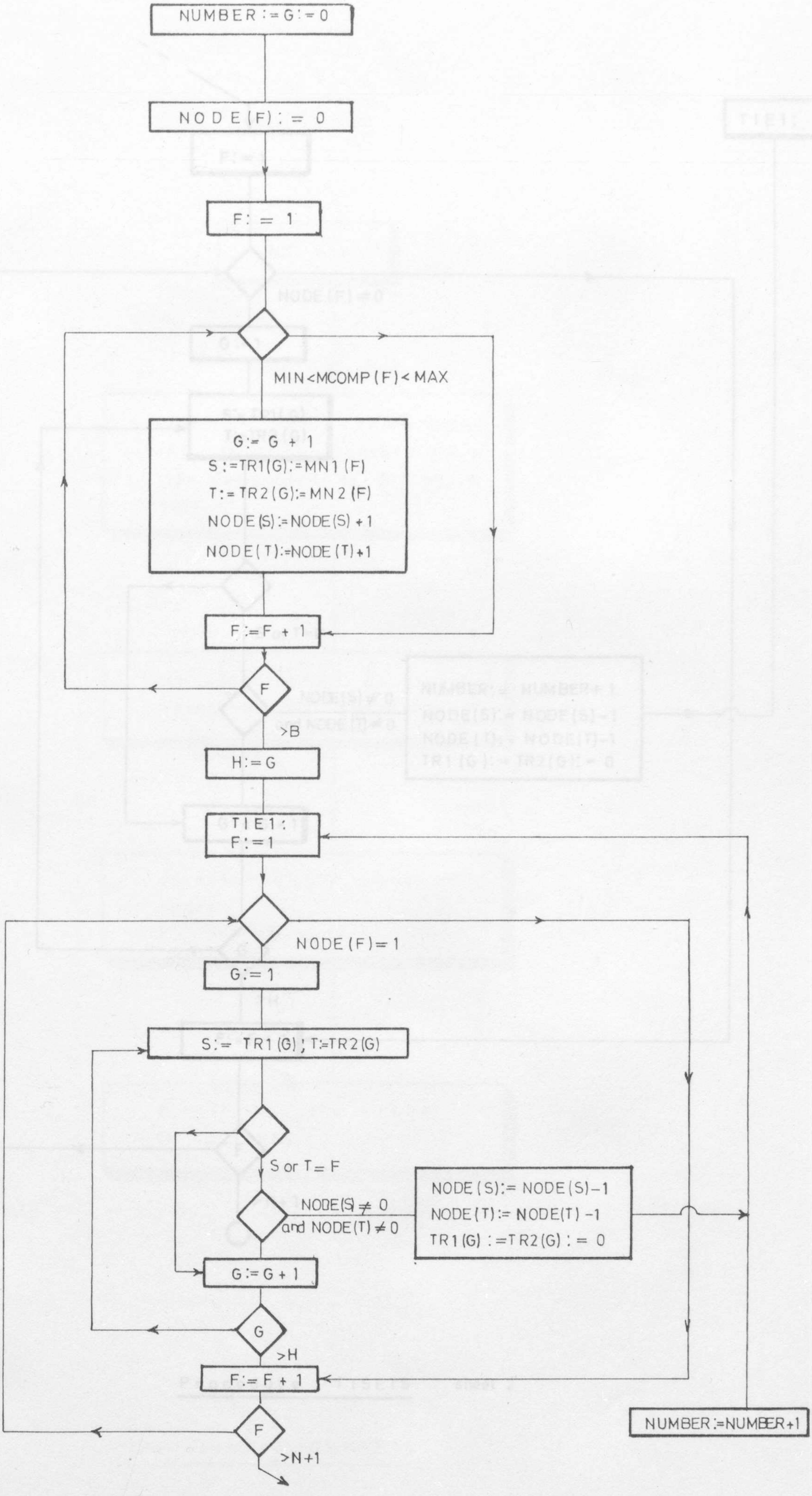
Program FOFSVMS


counts numbers of various types of constraints by procedures TISETS and CUTSETS
Calls procedures TREE and TIESET MATRIX
Reorders BRANCHES into TREE and LINKS
FORMS partitioned degenerate STATE MATRIX A
Formulates equations of constraint by calls to procedures TREE, TIESET MATRIX and SUBSYSTEM
Solves interactive subsystems of constraint one by one and substitutes for degenerate variables in main system MATRIX A and remaining subsystems
Extracts any remaining resistor equations from main system, solves and substitutes back into the remaining equations which then become the STATE and OUTPUT STATE equations
Solves the STATE EQUATIONS using CERTAINE'S METHOD

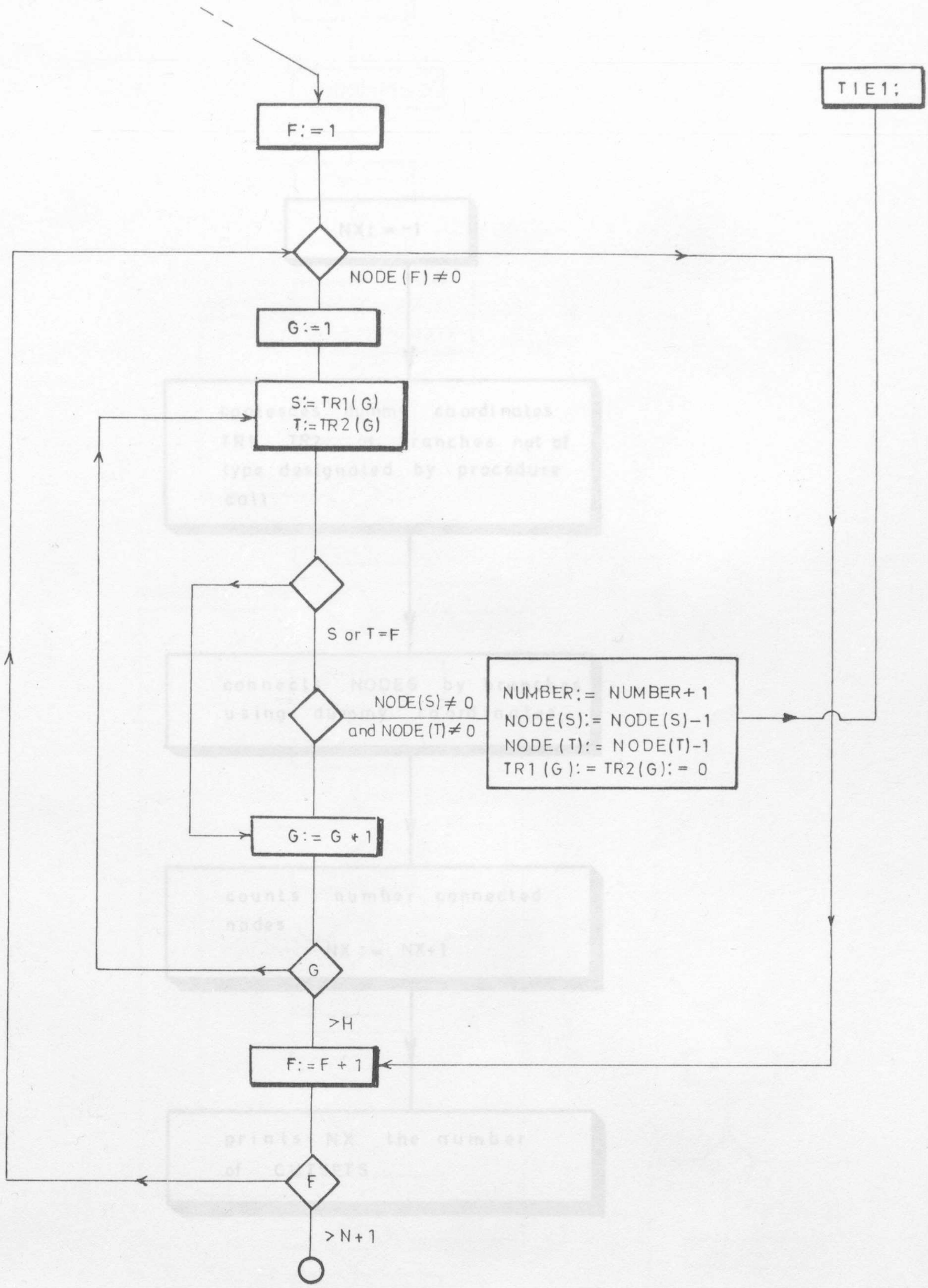
Program FOFSVMS


NUMBER:=∅
CONNECTS NODES by type of branches designated by procedure call
TIE1: Takes out any dangling branches
are there any branches still connecting nodes ?
YES
takes out a branch NUMBER:=NUMBER+1
NO
PRINT NUMBER OF TIESETS

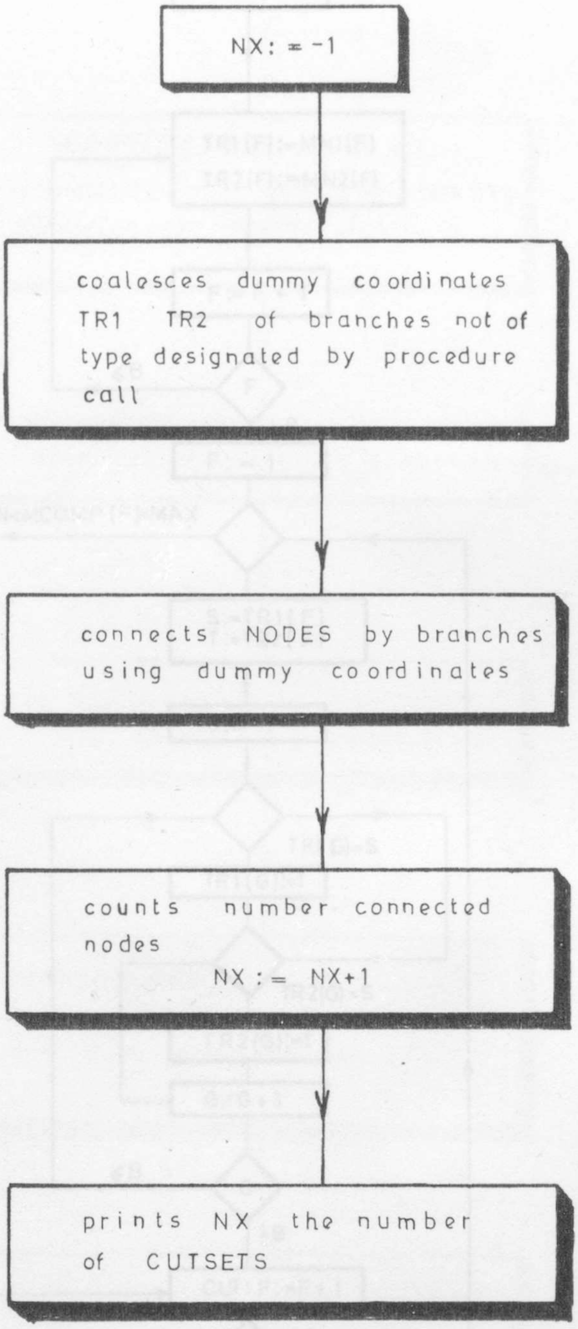
Procedure TISETS



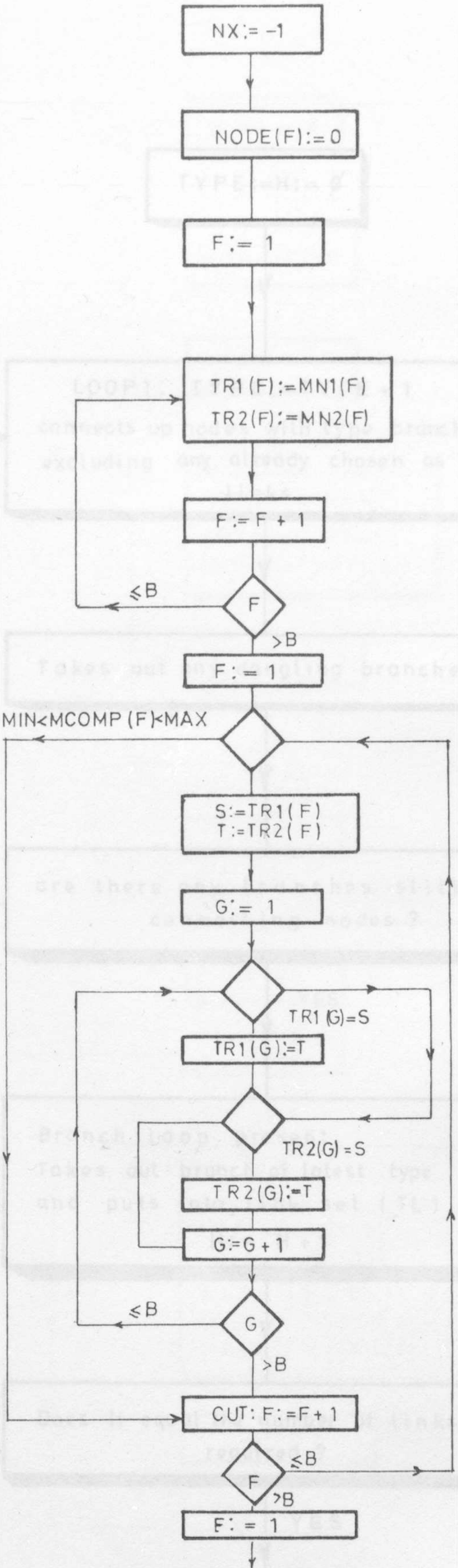
Procedure TISETS sheet 1

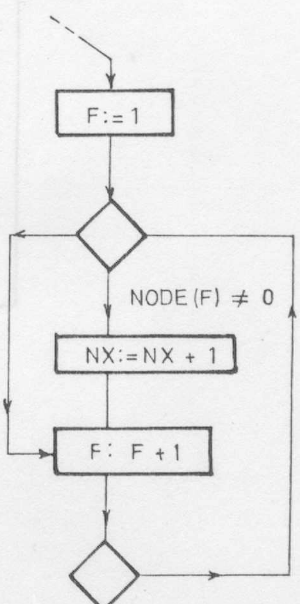


Procedure TISETS sheet 2

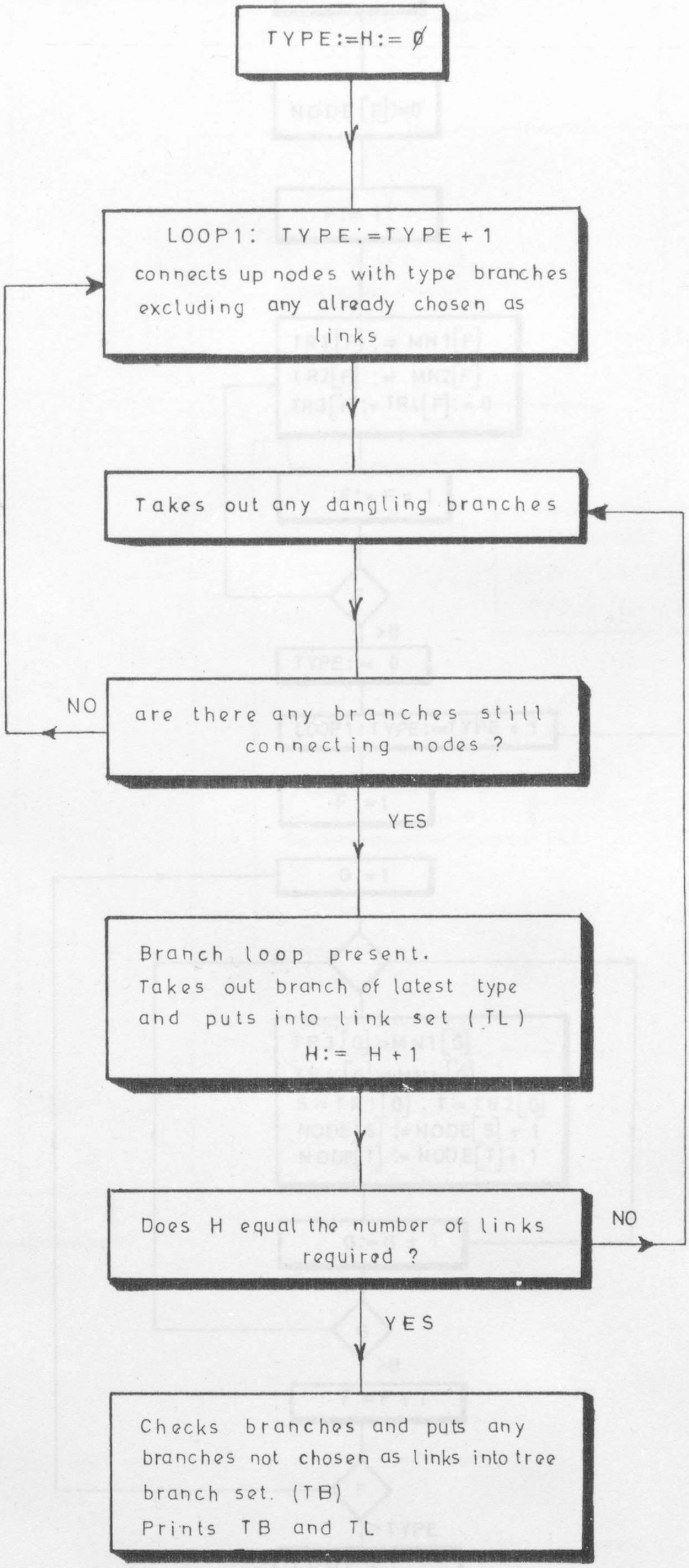

NX: = -1
coalesces dummy coordinates TR1 TR2 of branches not of type designated by procedure call
connects NODES by branches using dummy coordinates
counts number connected nodes
NX := NX+1
prints NX the number of CUTSETS

Procedure CUTSETS

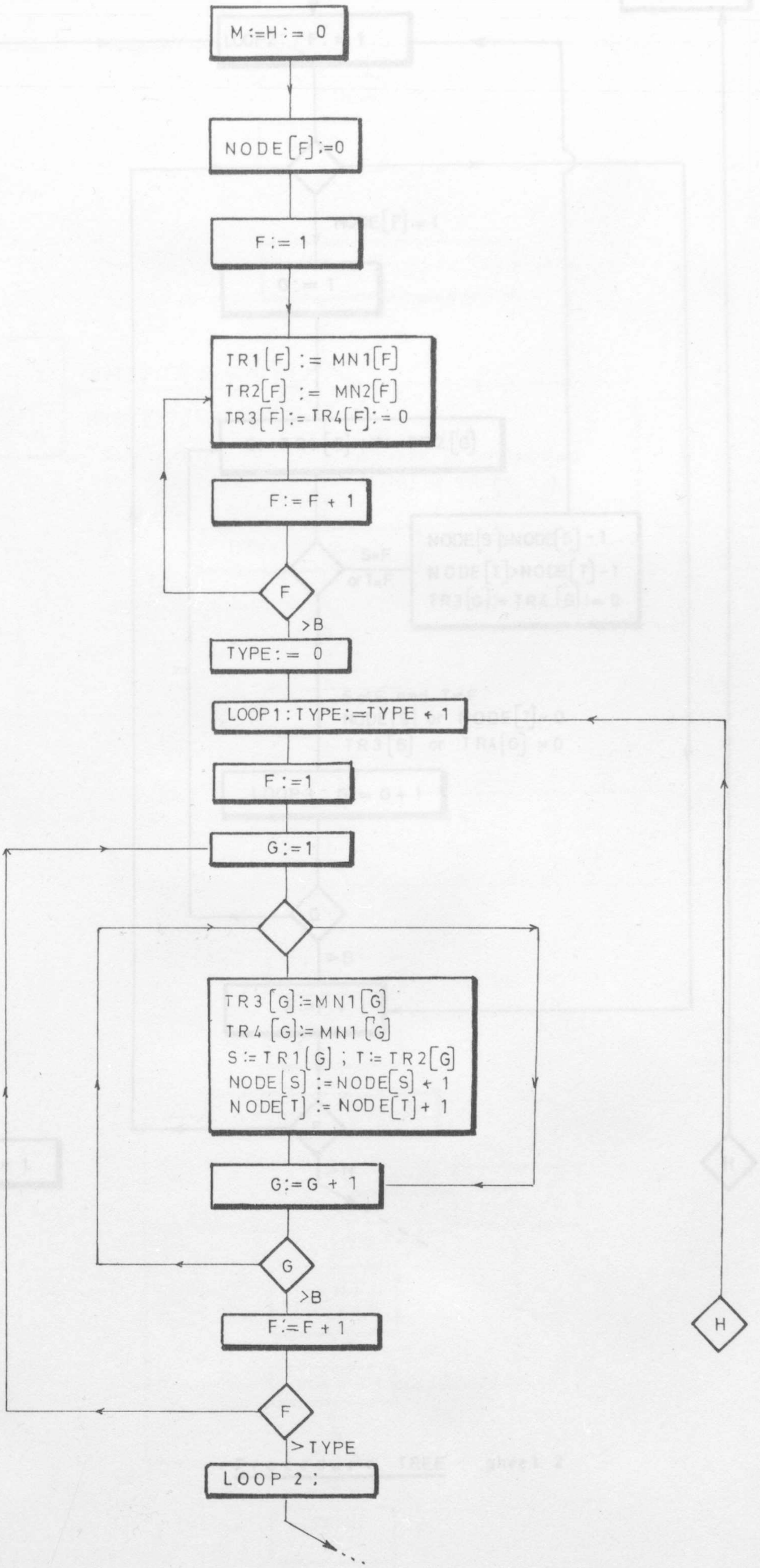




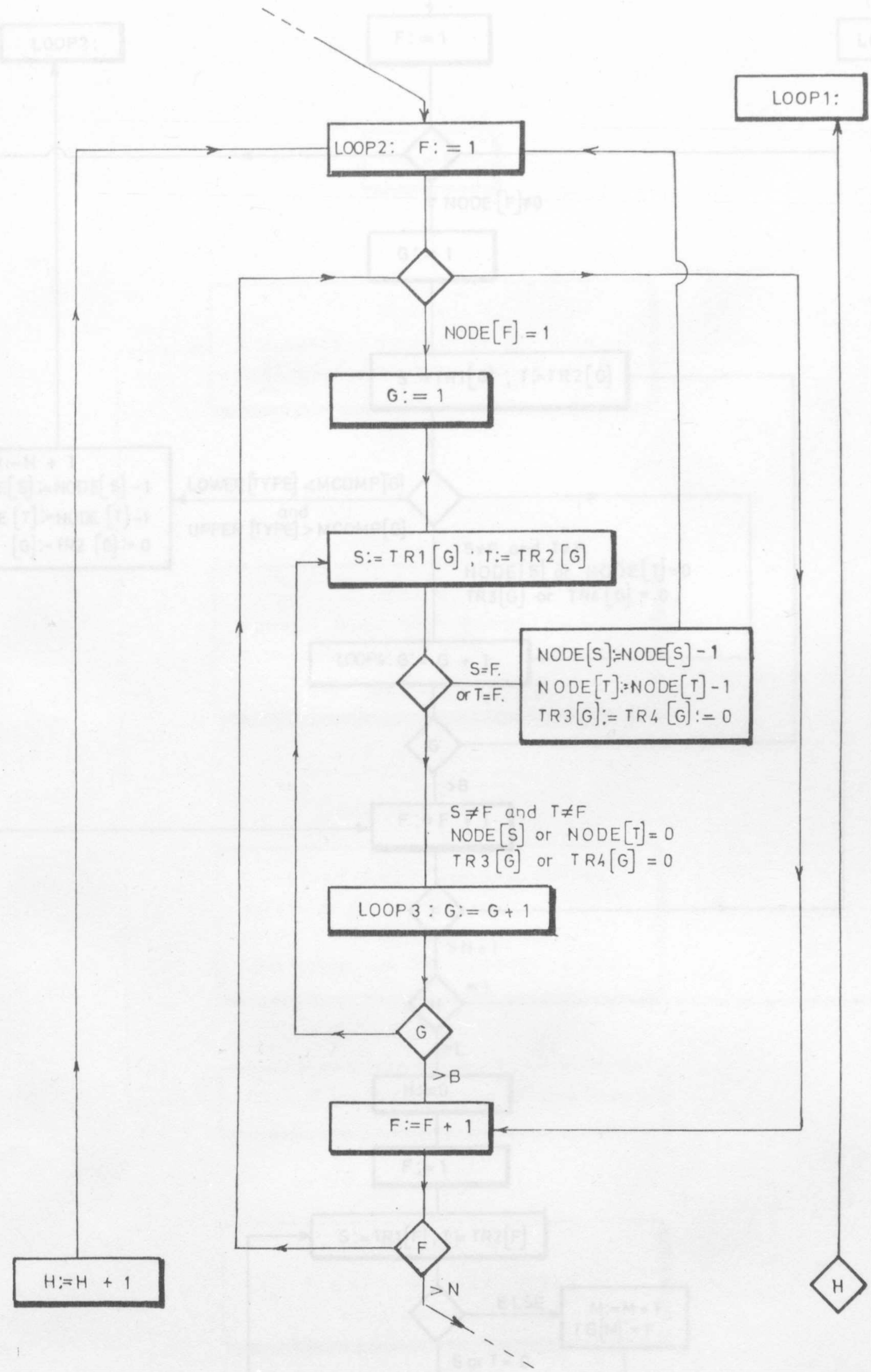
Procedure CUTSETS


TYPE:=H:= Ø
LOOP1: TYPE:=TYPE + 1
connects up nodes with type branches excluding any already chosen as links
Takes out any dangling branches
NO
are there any branches still connecting nodes ?
YES
Branch loop present.
Takes out branch of latest type and puts into link set (TL)
H:= H + 1
Does H equal the number of links required ?
NO
YES
Checks branches and puts any branches not chosen as links into tree branch set. (TB)
Prints TB and TL

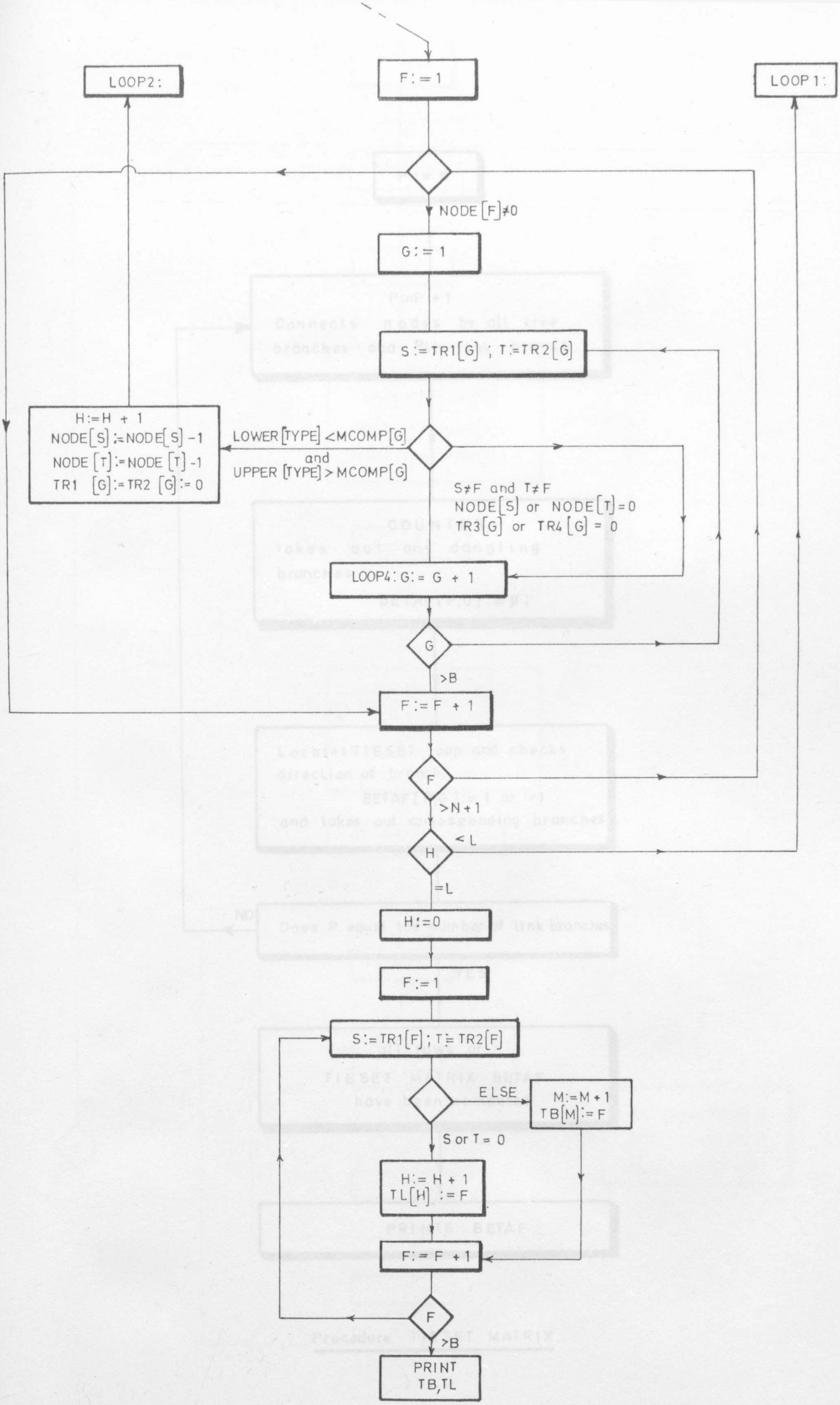
Procedure TREE



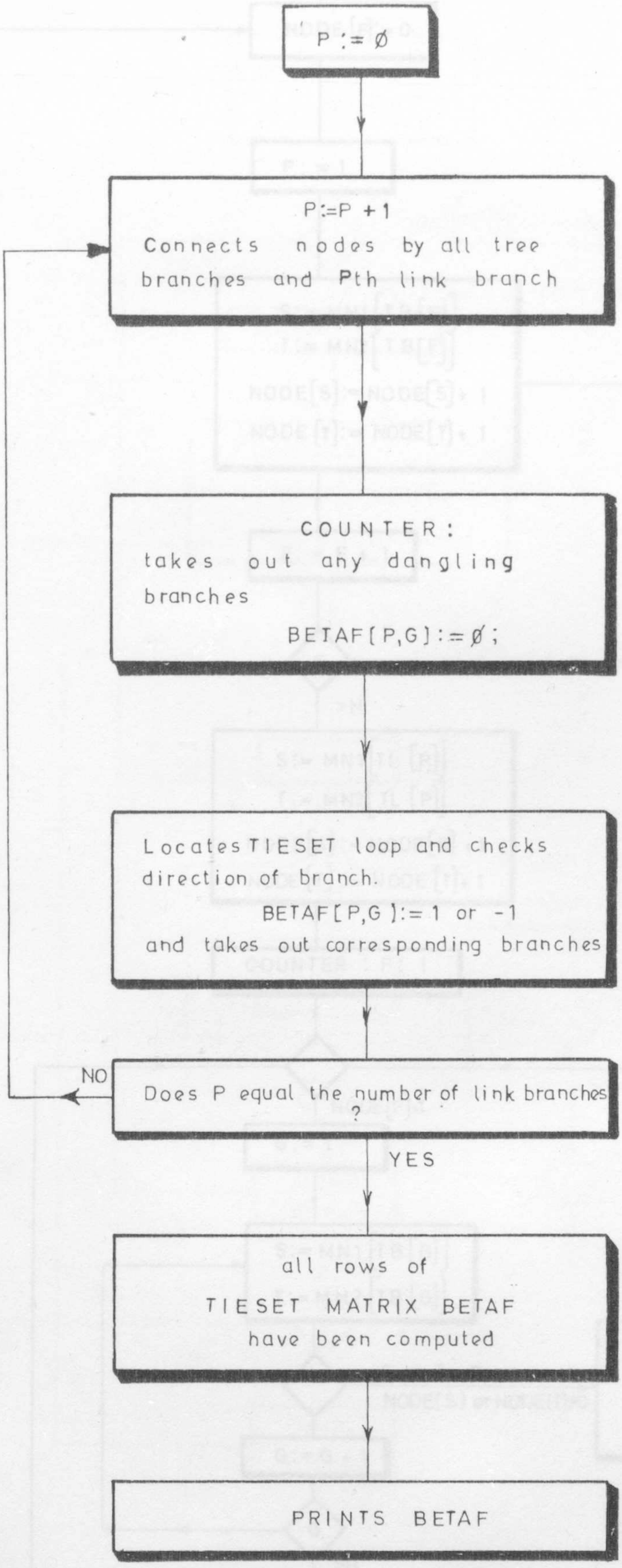
Procedure TREE sheet 1



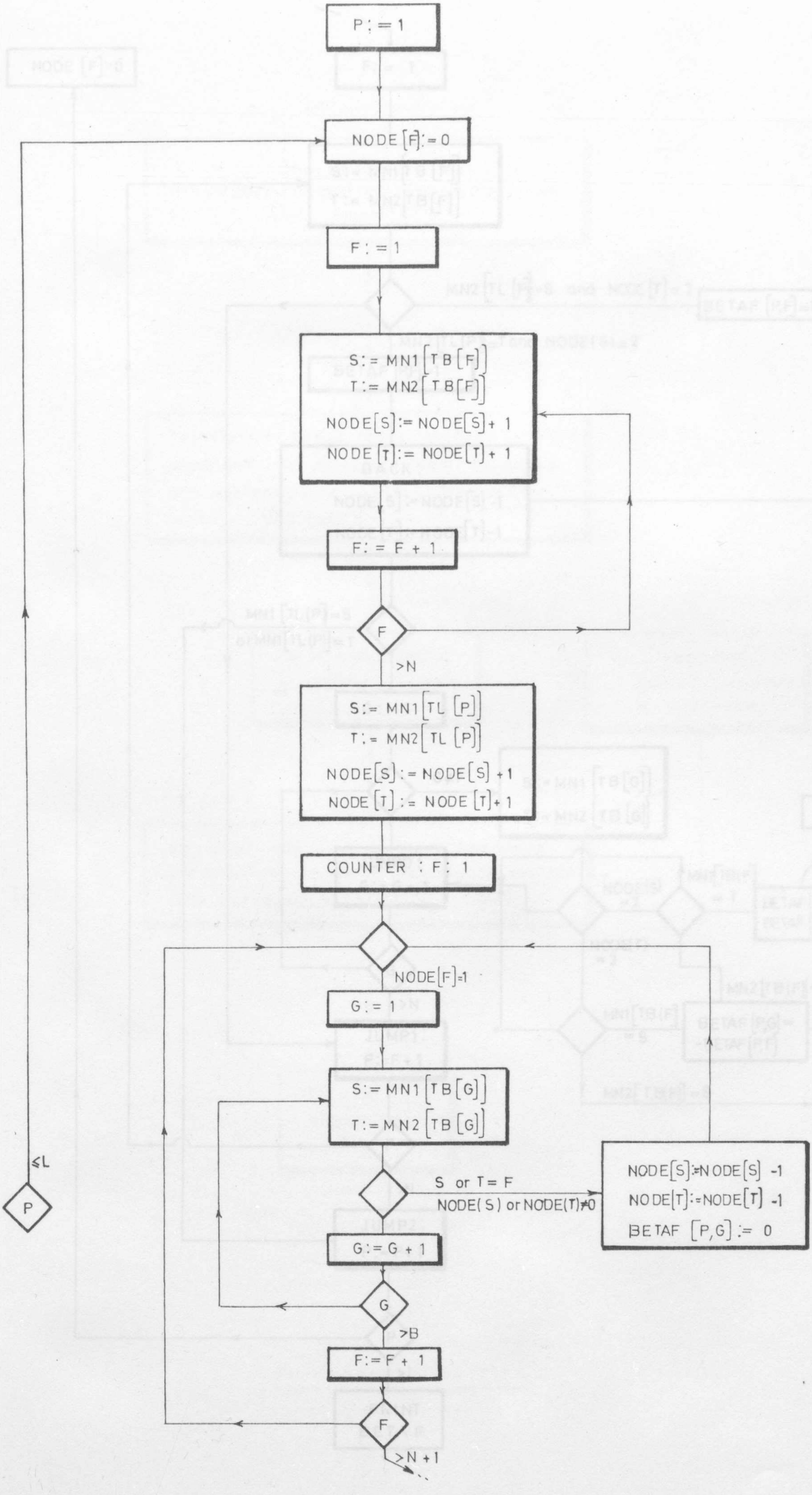
Procedure TREE sheet 2



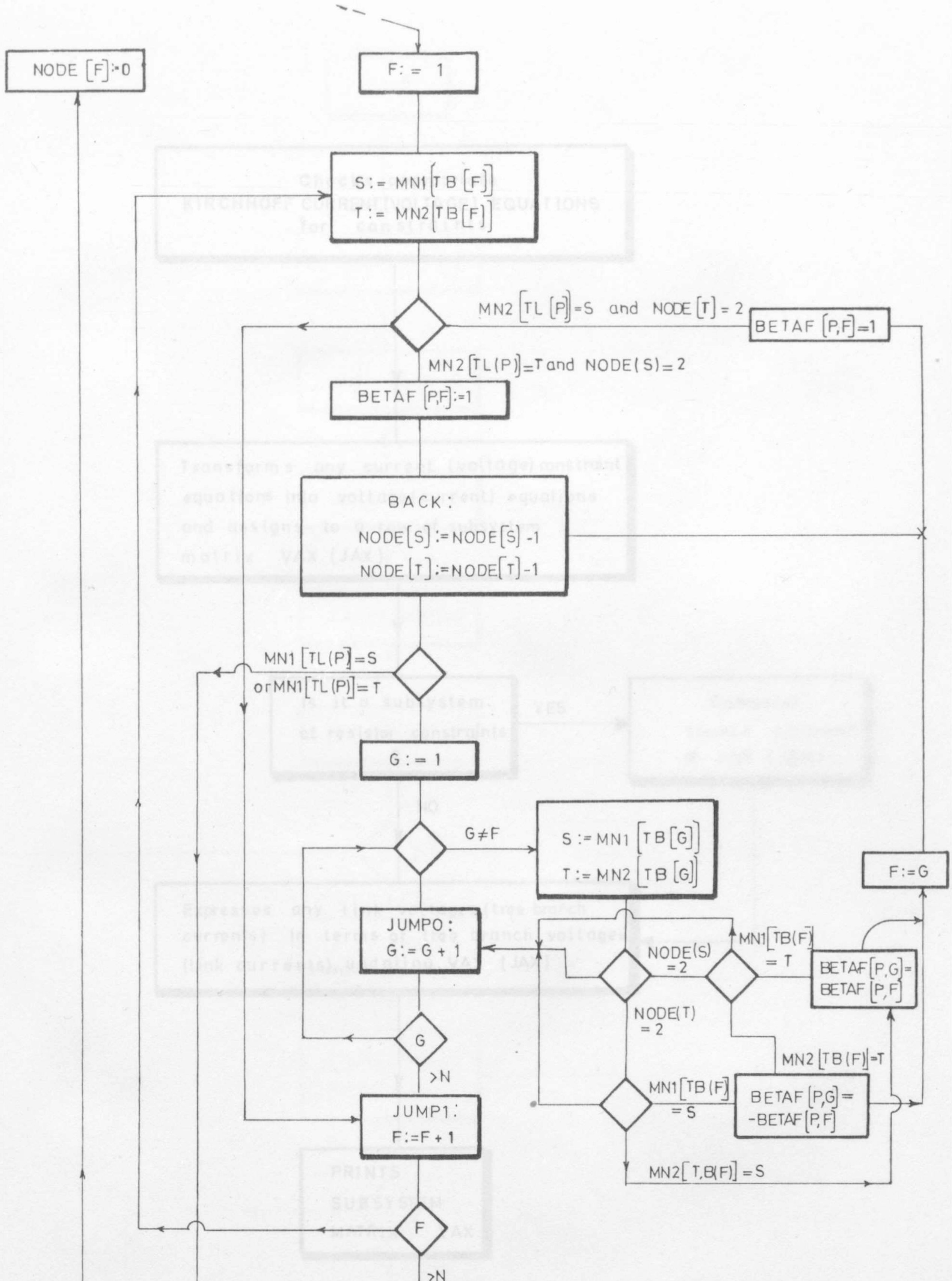
Procedure TREE sheet 3


P := ∅
P:=P +1
Connects nodes by all tree branches and Pth link branch
COUNTER:
takes out any dangling branches
BETAF[P,G]:=∅;
Locates TIESET loop and checks direction of branch.
BETAF[P,G]:= 1 or -1
and takes out corresponding branches
NO
Does P equal the number of link branches ?
YES
all rows of
TIESET MATRIX BETAF
have been computed
PRINTS BETAF

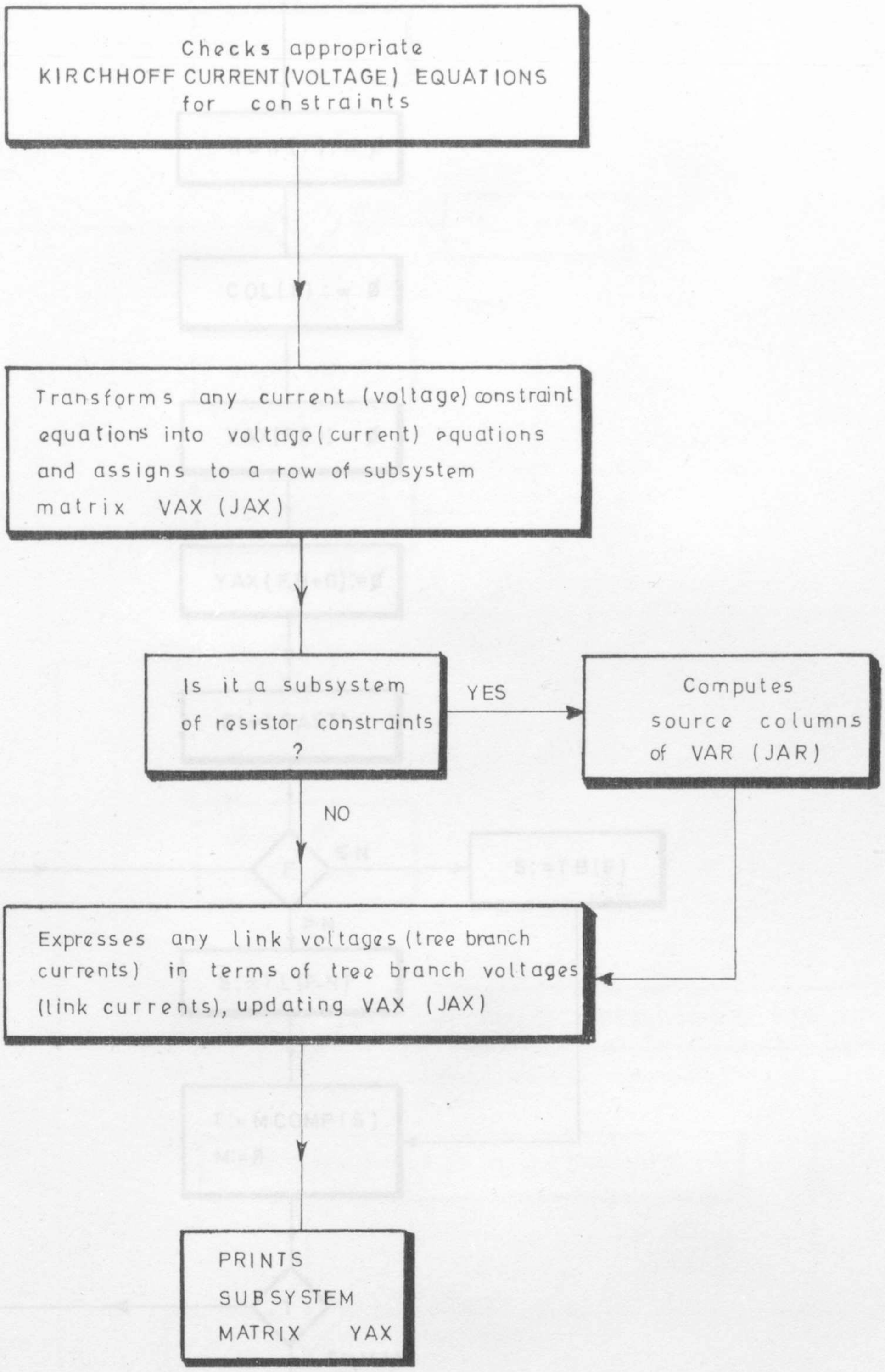
Procedure TIESET MATRIX



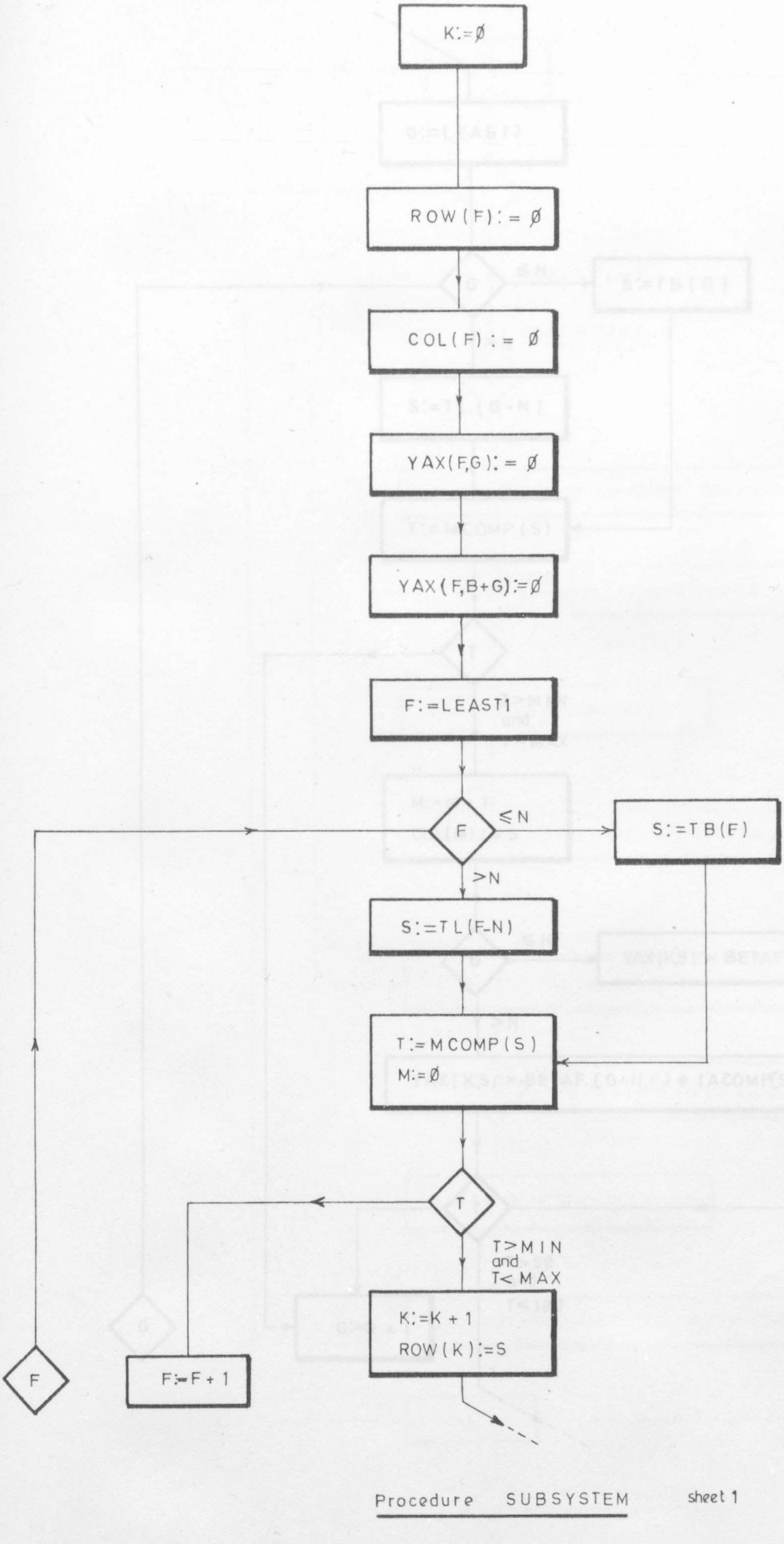
Procedure TIESET MATRIX sheet 1



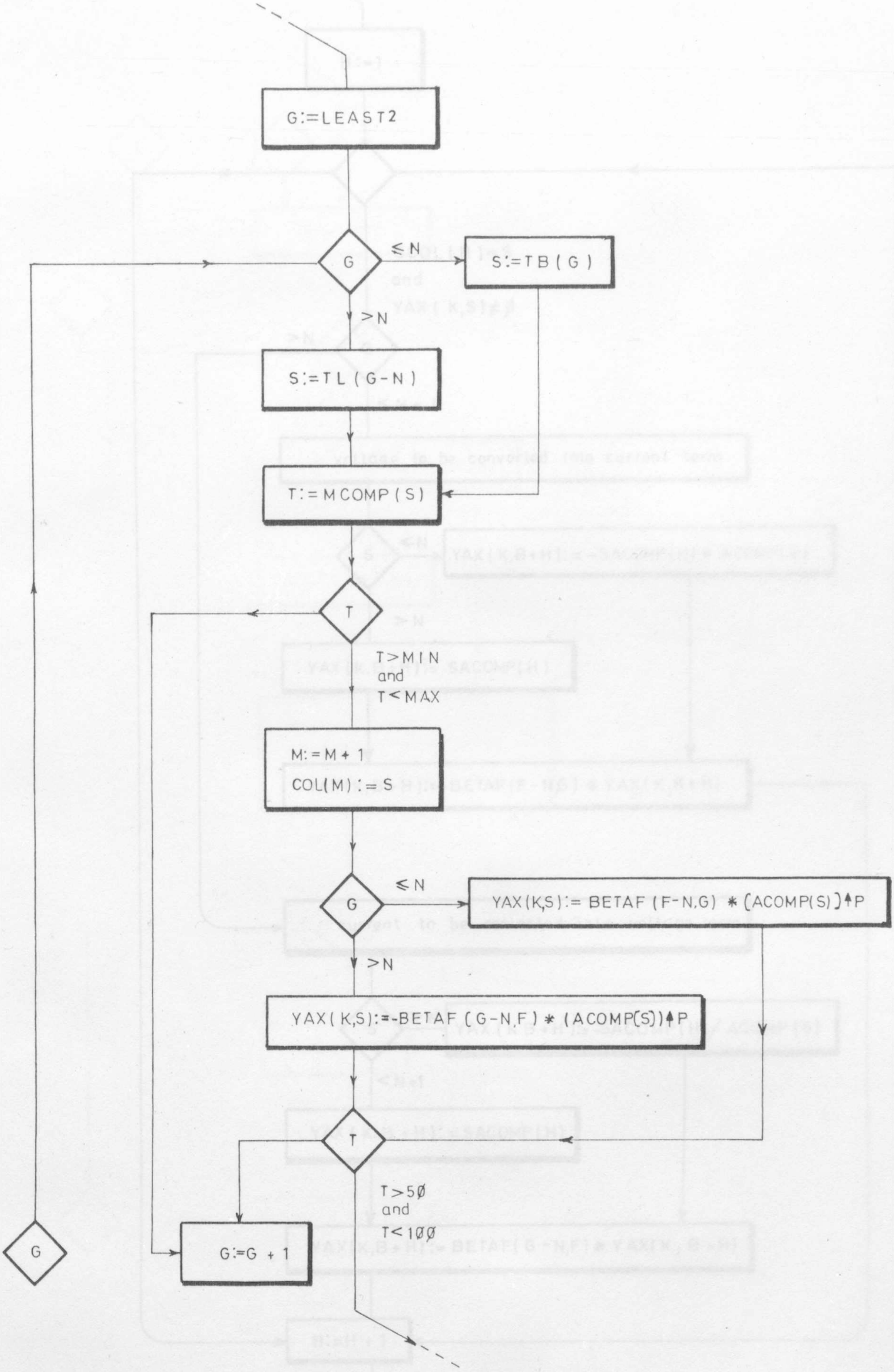
Procedure TIESET MATRIX sheet 2


Checks appropriate KIRCHHOFF CURRENT (VOLTAGE) EQUATIONS for constraints
Transforms any current (voltage) constraint equations into voltage (current) equations and assigns to a row of subsystem matrix VAX (JAX)
Is it a subsystem of resistor constraints ?
YES
Computes source columns of VAR (JAR)
NO
Expresses any link voltages (tree branch currents) in terms of tree branch voltages (link currents), updating VAX (JAX)
PRINTS SUBSYSTEM MATRIX YAX

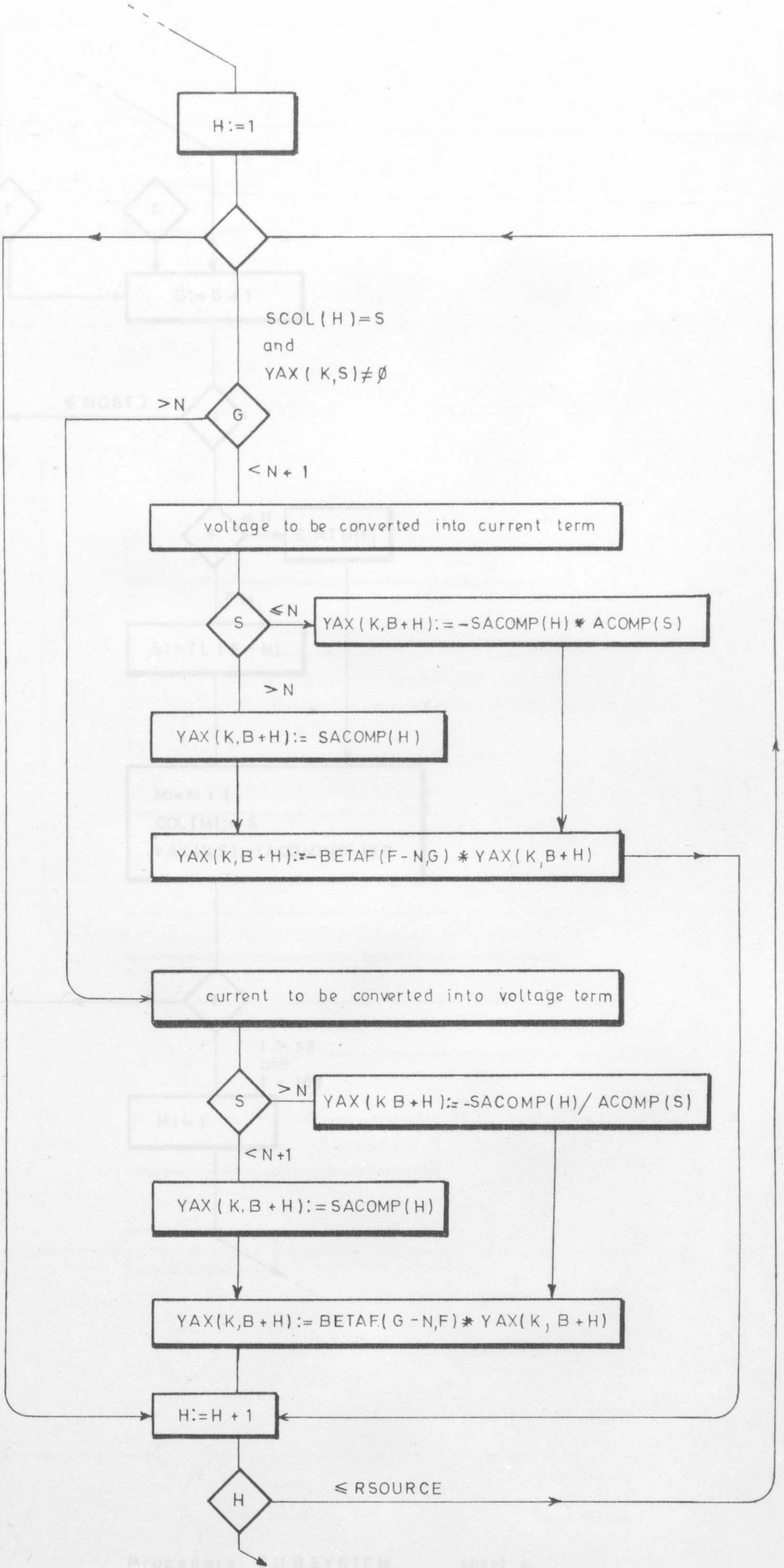
Procedure SUBSYSTEM



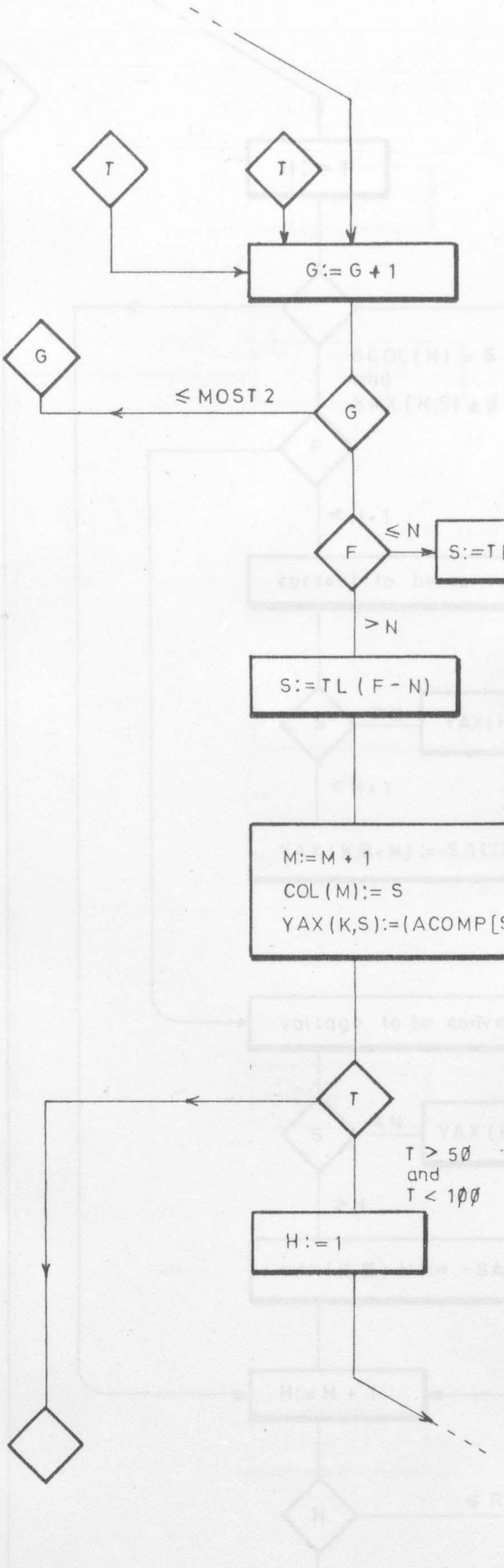
Procedure SUBSYSTEM sheet 1



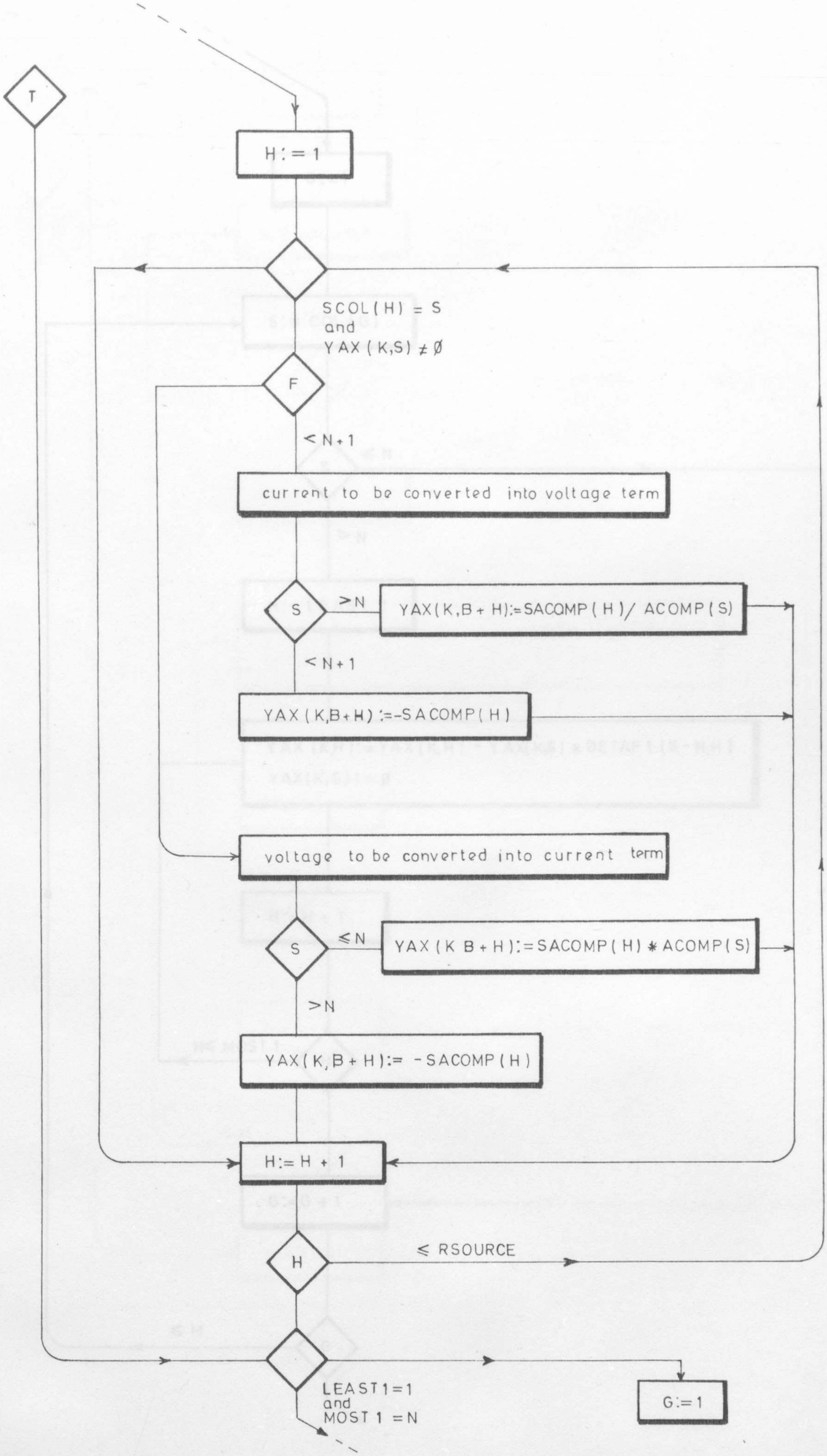
Procedure SUBSYSTEM sheet 2


H := 1
SCOL(H) = S
and
YAX(K,S) ≠ ∅
> N
G
< N + 1
voltage to be converted into current term
S
≤ N
YAX(K,B+H) := −SACOMP(H) * ACOMP(S)
> N
YAX(K,B+H) := SACOMP(H)
YAX(K,B+H) := −BETAF(F−N,G) * YAX(K,B+H)
current to be converted into voltage term
S
> N
YAX(K B+H) := −SACOMP(H) / ACOMP(S)
< N+1
YAX(K, B+H) := SACOMP(H)
YAX(K,B+H) := BETAF(G−N,F) * YAX(K, B+H)
H := H + 1
H
≤ RSOURCE

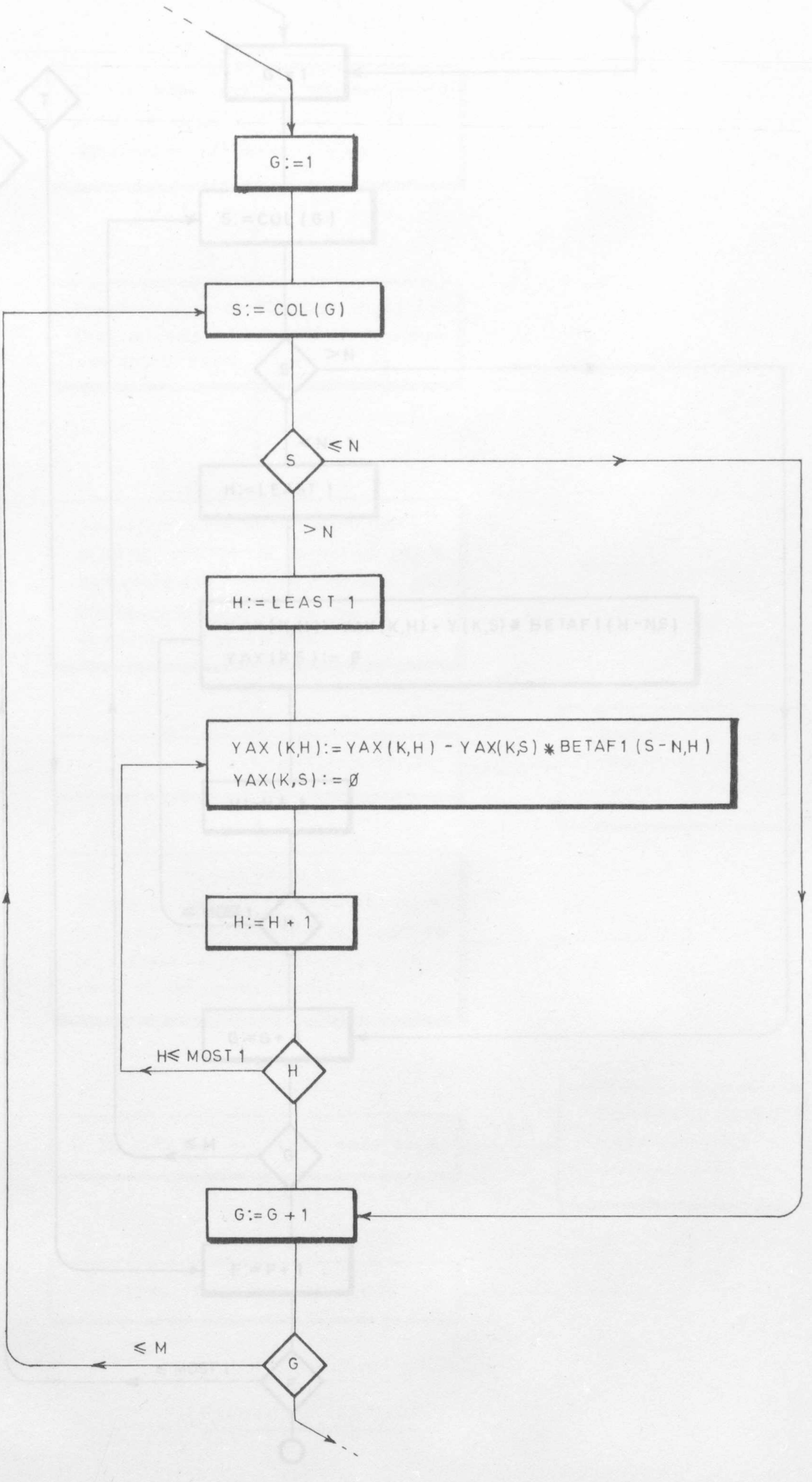
Procedure SUBSYSTEM sheet 3



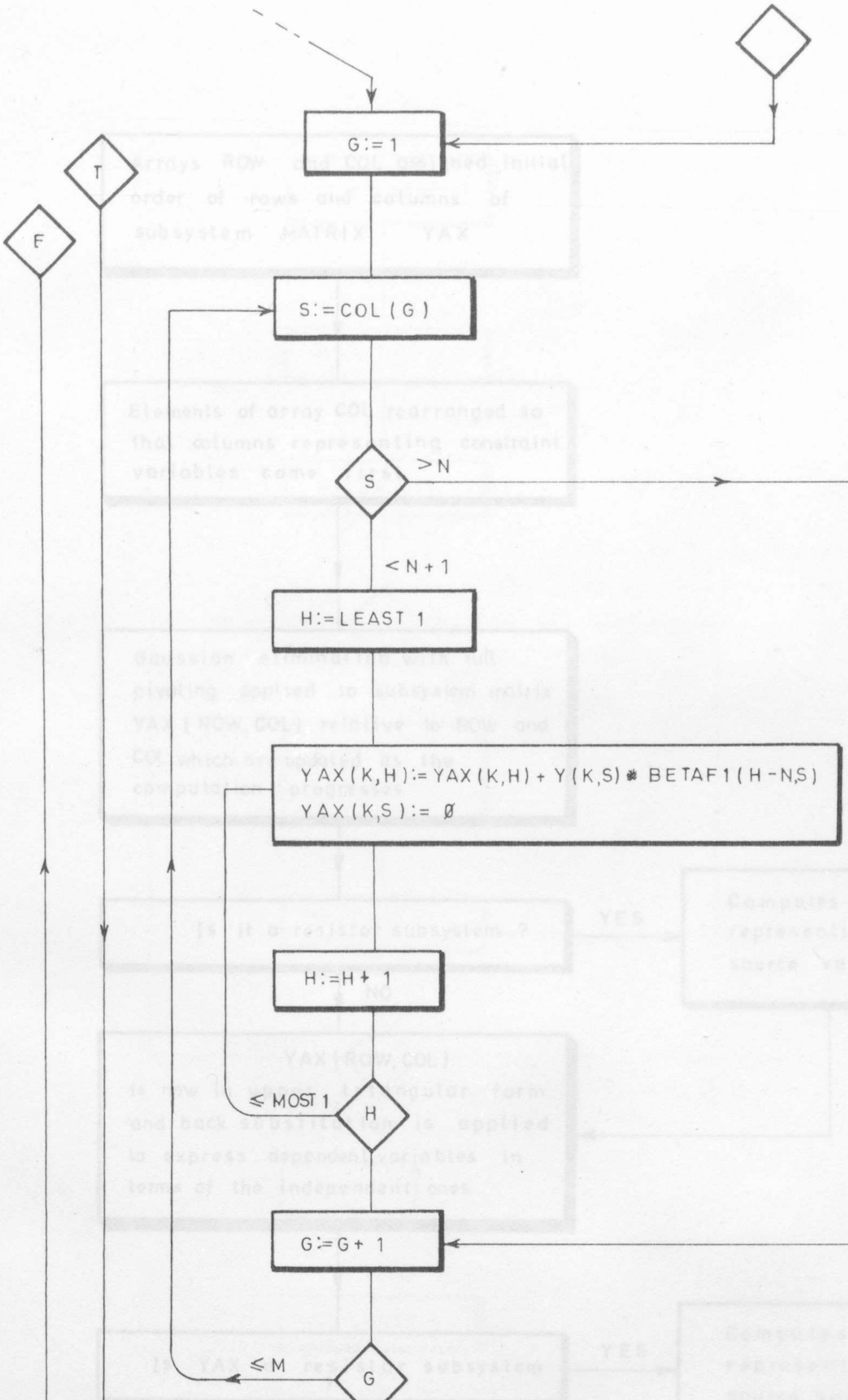
Procedure SUBSYSTEM sheet 4


T
H:= 1
SCOL(H) = S
and
YAX (K,S) ≠ Ø
F
< N+1
current to be converted into voltage term
S
> N
YAX(K,B + H):=SACOMP(H)/ ACOMP(S)
< N+1
YAX(K,B+H):=-SACOMP(H)
voltage to be converted into current term
S
≤ N
YAX(K B + H):=SACOMP(H) * ACOMP(S)
> N
YAX(K,B + H):= -SACOMP(H)
H:= H + 1
H
≤ RSOURCE
LEAST1=1
and
MOST1 = N
G:= 1

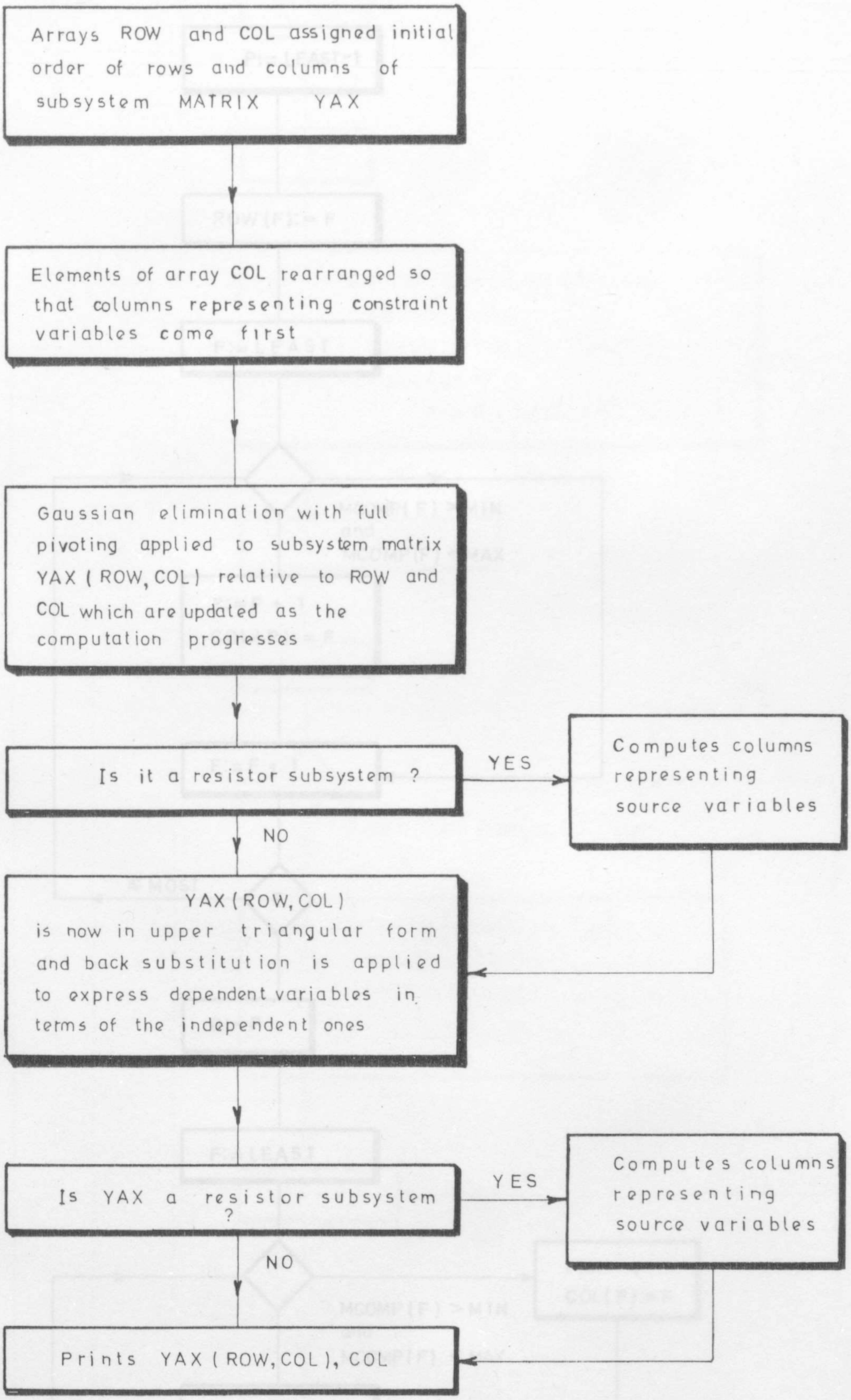
Procedure SUBSYSTEM sheet 5



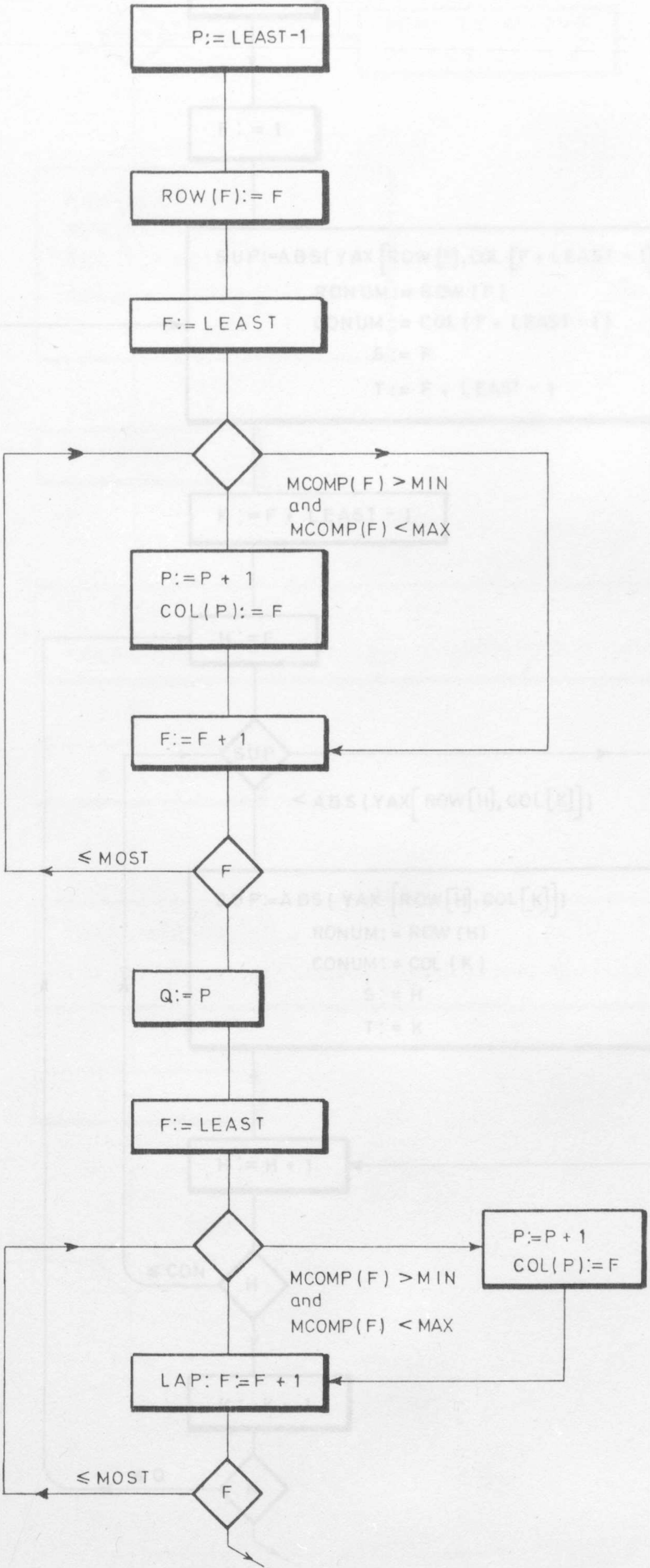
Procedure SUBSYSTEM sheet 6



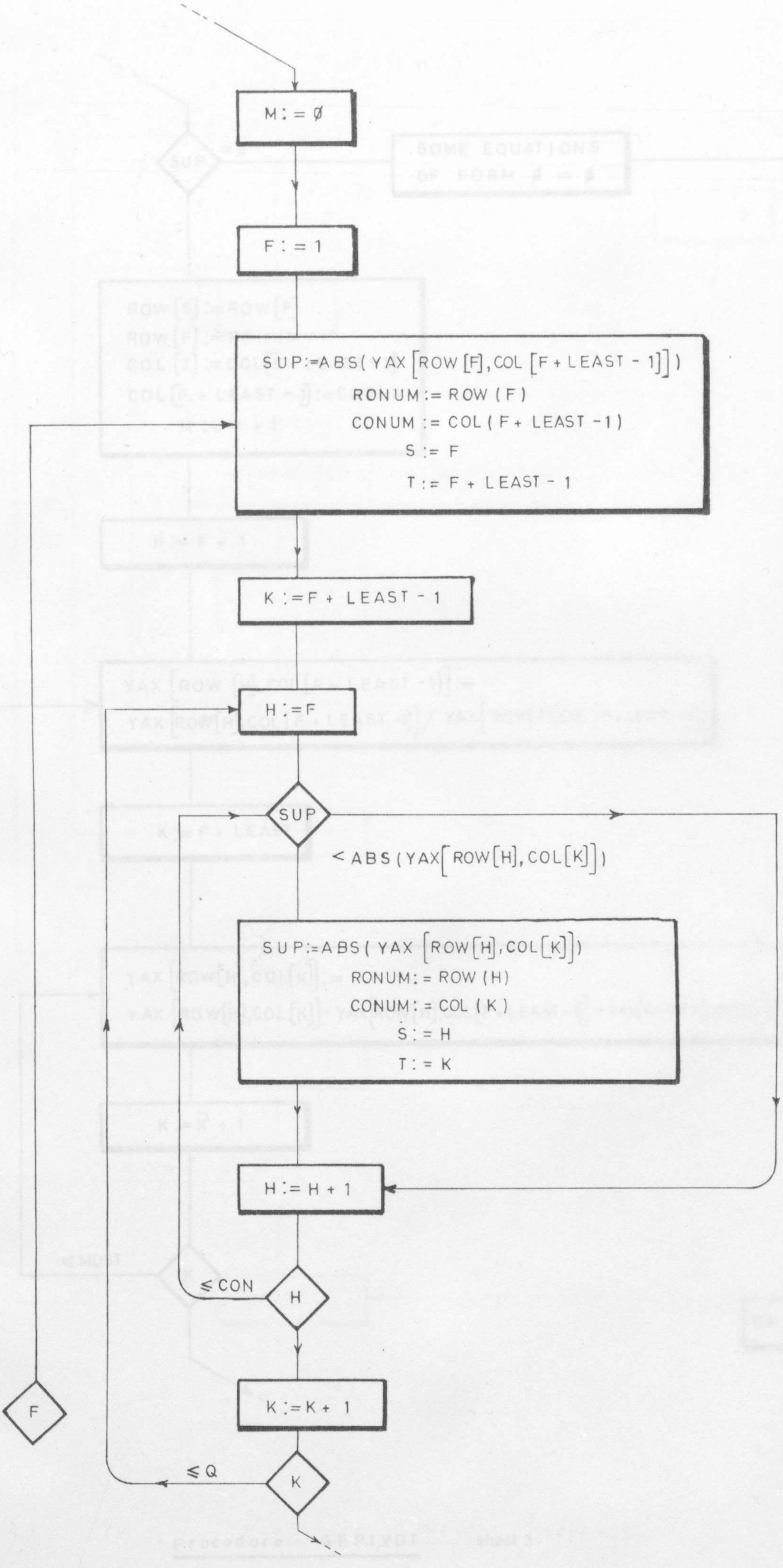
Procedure SUBSYSTEM sheet 7


Arrays ROW and COL assigned initial order of rows and columns of subsystem MATRIX YAX
Elements of array COL rearranged so that columns representing constraint variables come first
Gaussian elimination with full pivoting applied to subsystem matrix YAX (ROW, COL) relative to ROW and COL which are updated as the computation progresses
Is it a resistor subsystem ?
YES
Computes columns representing source variables
NO
YAX (ROW, COL) is now in upper triangular form and back substitution is applied to express dependent variables in terms of the independent ones
Is YAX a resistor subsystem ?
YES
Computes columns representing source variables
NO
Prints YAX (ROW, COL), COL

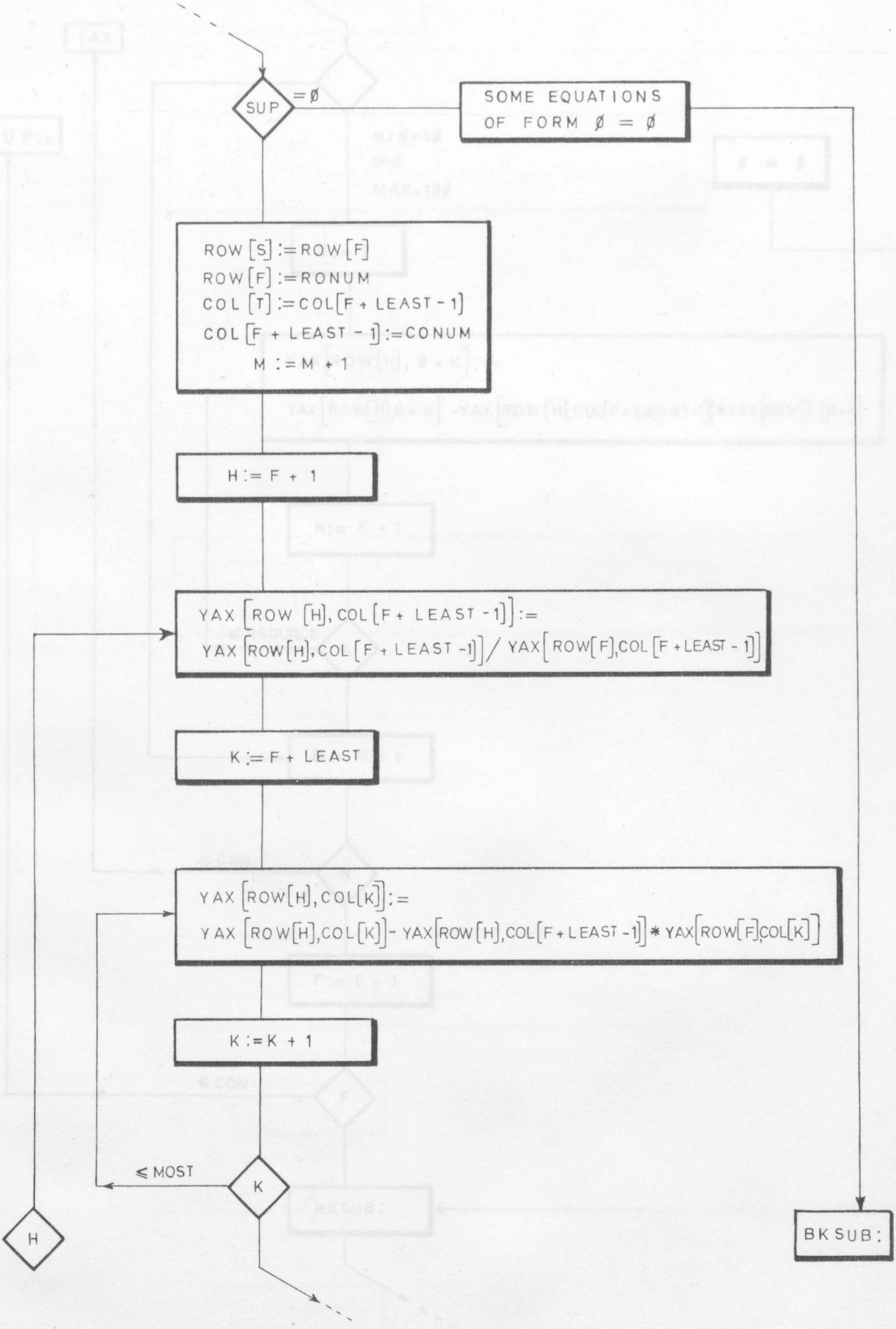
Procedure GFPIVOT



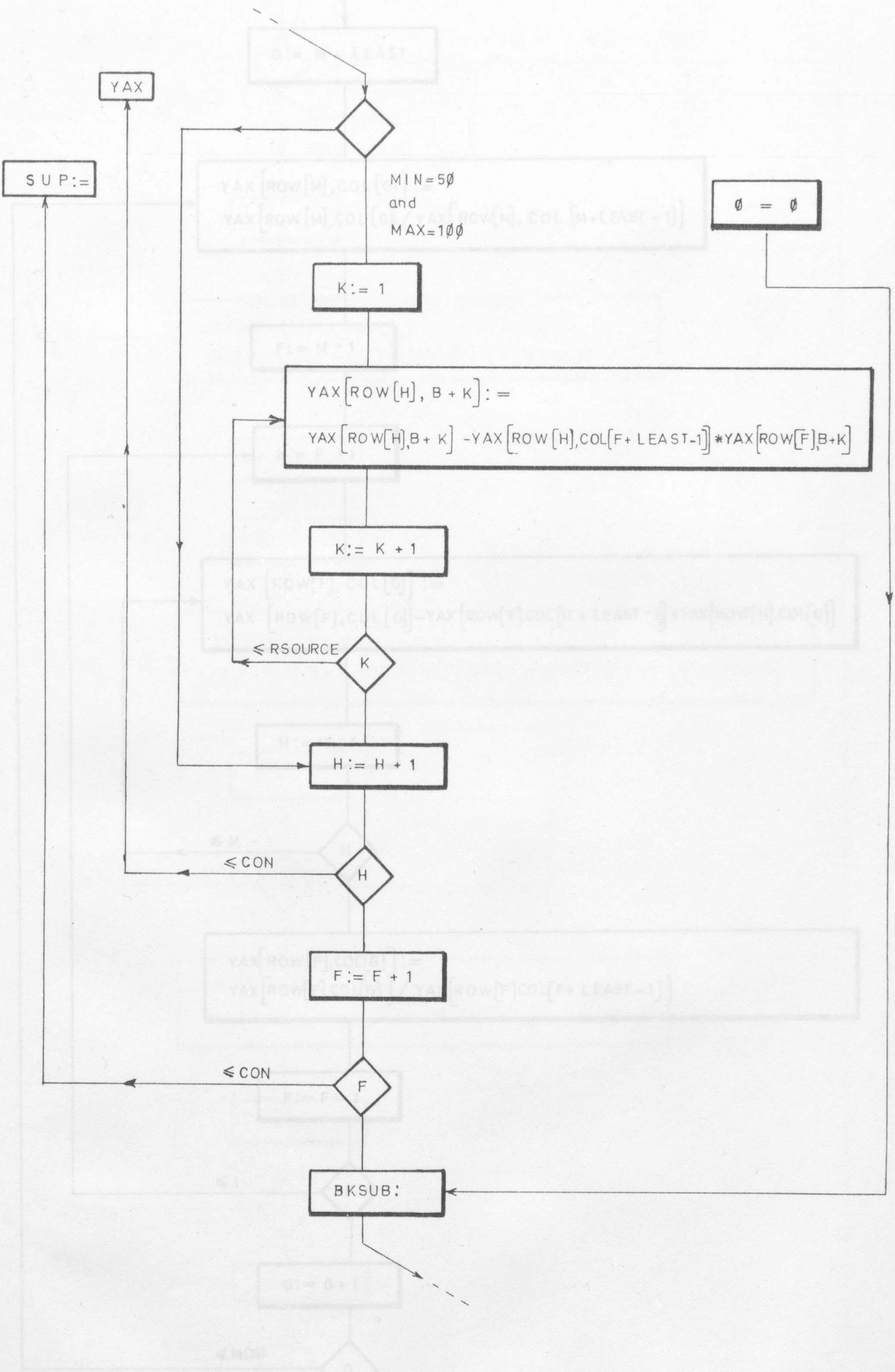
Procedure GFPIVOT sheet 1



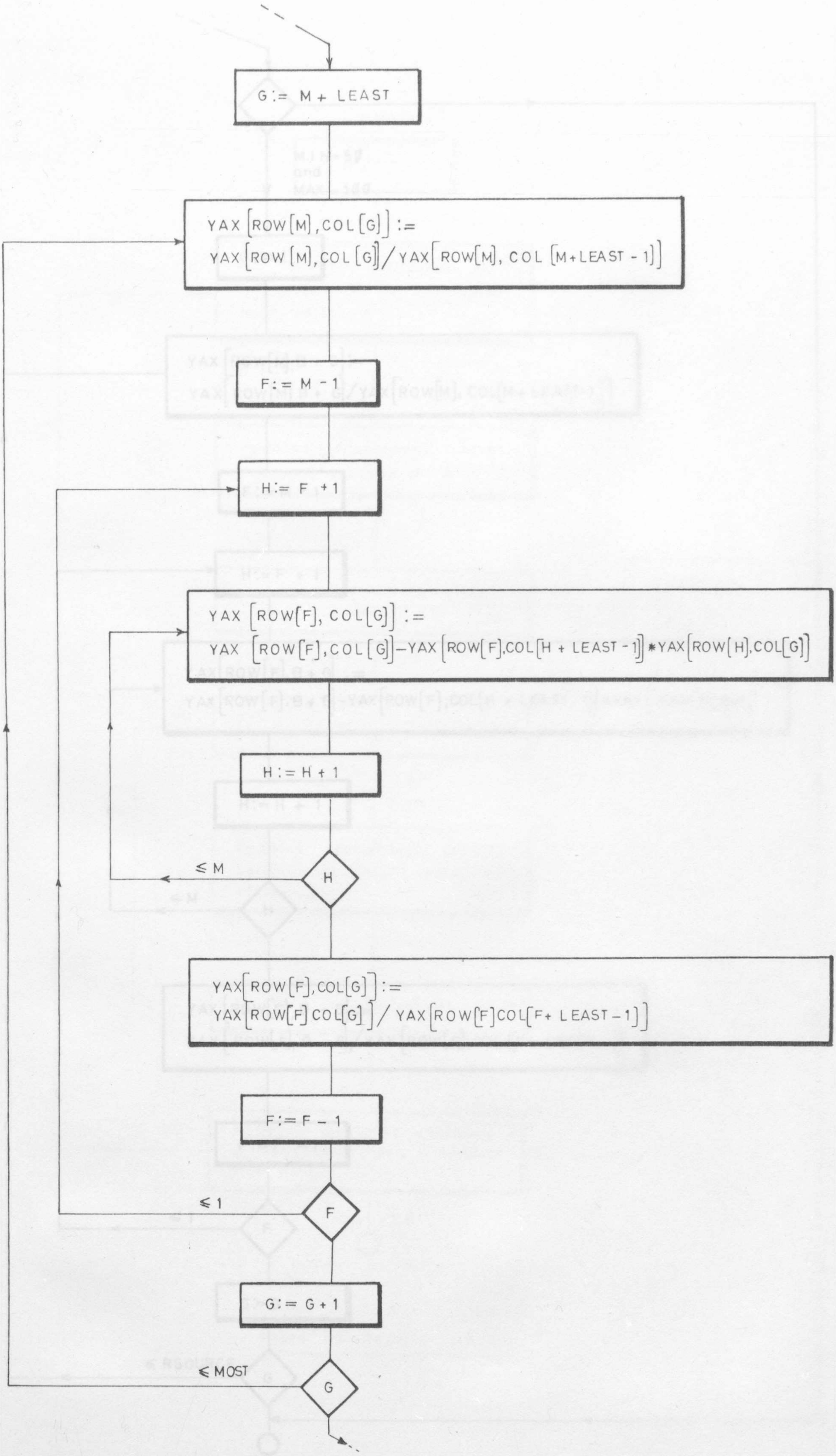
Procedure GFPIVOT sheet 2


SUP
= Ø
SOME EQUATIONS OF FORM Ø = Ø
ROW [S] := ROW [F]
ROW [F] := RONUM
COL [T] := COL [F + LEAST - 1]
COL [F + LEAST - 1] := CONUM
M := M + 1
H := F + 1
YAX [ROW [H], COL [F + LEAST - 1]] :=
YAX [ROW [H], COL [F + LEAST - 1]] / YAX [ROW [F], COL [F + LEAST - 1]]
K := F + LEAST
YAX [ROW [H], COL [K]] :=
YAX [ROW [H], COL [K]] - YAX [ROW [H], COL [F + LEAST - 1]] * YAX [ROW [F], COL [K]]
K := K + 1
≤ MOST
K
H
BKSUB:

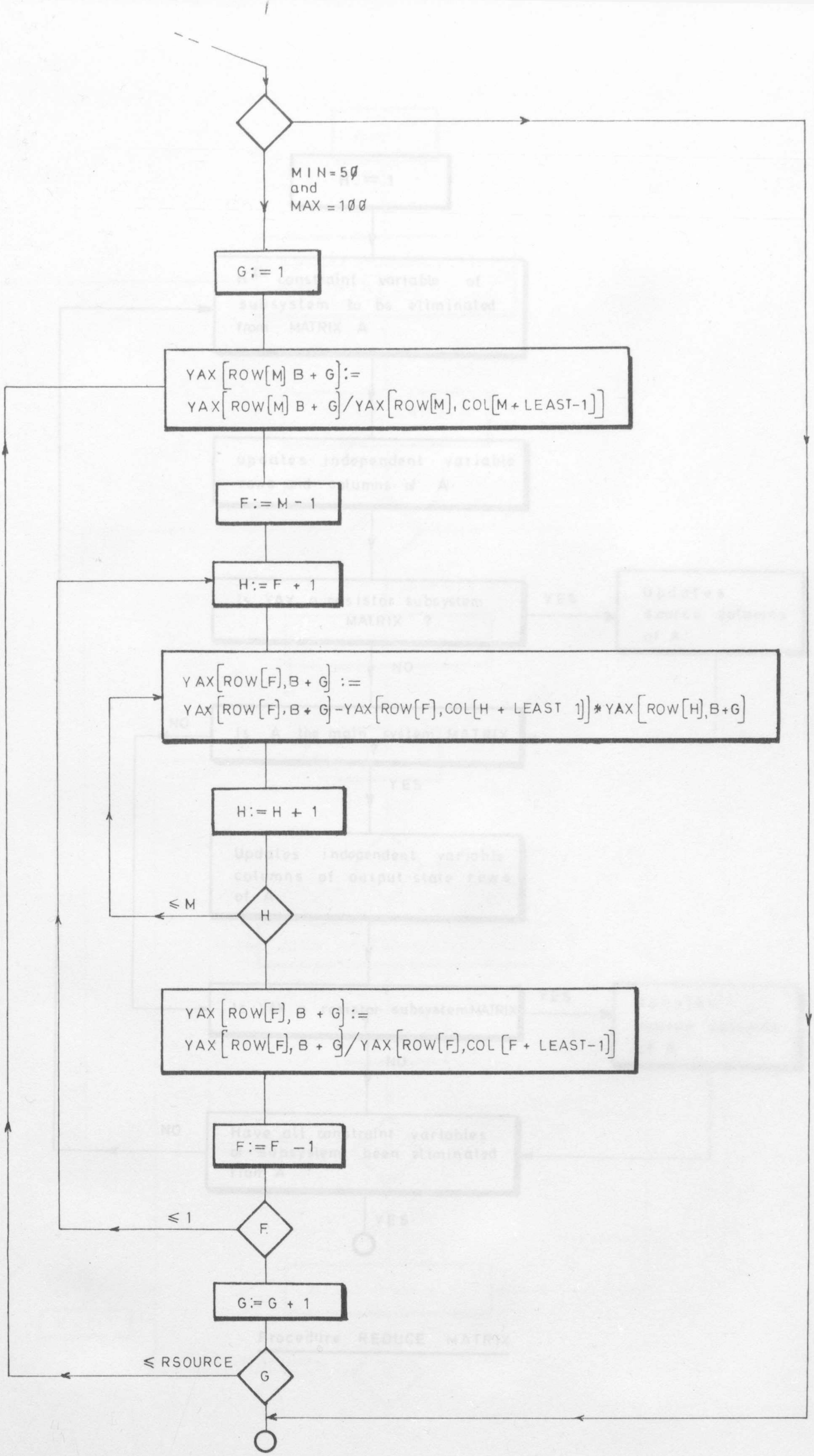
Procedure GFPIVOT sheet 3



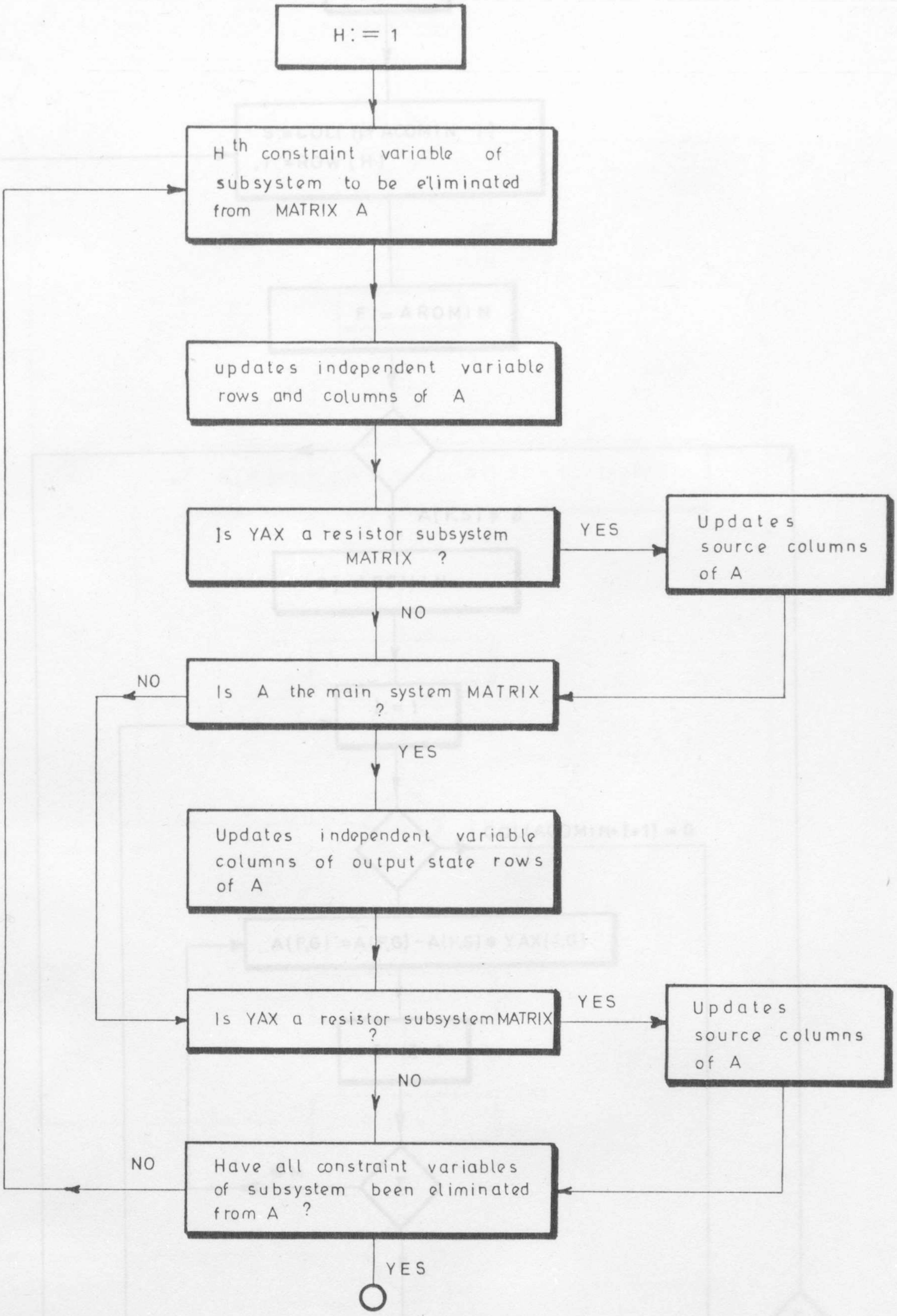
Procedure GFPIVOT sheet 4


G := M + LEAST
YAX [ROW [M], COL [G]] :=
YAX [ROW [M], COL [G]] / YAX [ROW [M], COL [M+LEAST - 1]]
F := M - 1
H := F + 1
YAX [ROW [F], COL [G]] :=
YAX [ROW [F], COL [G] - YAX [ROW [F], COL [H + LEAST - 1]] * YAX [ROW [H], COL [G]]
H := H + 1
≤ M
H
YAX [ROW [F], COL [G]] :=
YAX [ROW [F] COL [G]] / YAX [ROW [F] COL [F+ LEAST - 1]]
F := F - 1
≤ 1
F
G := G + 1
≤ MOST
G

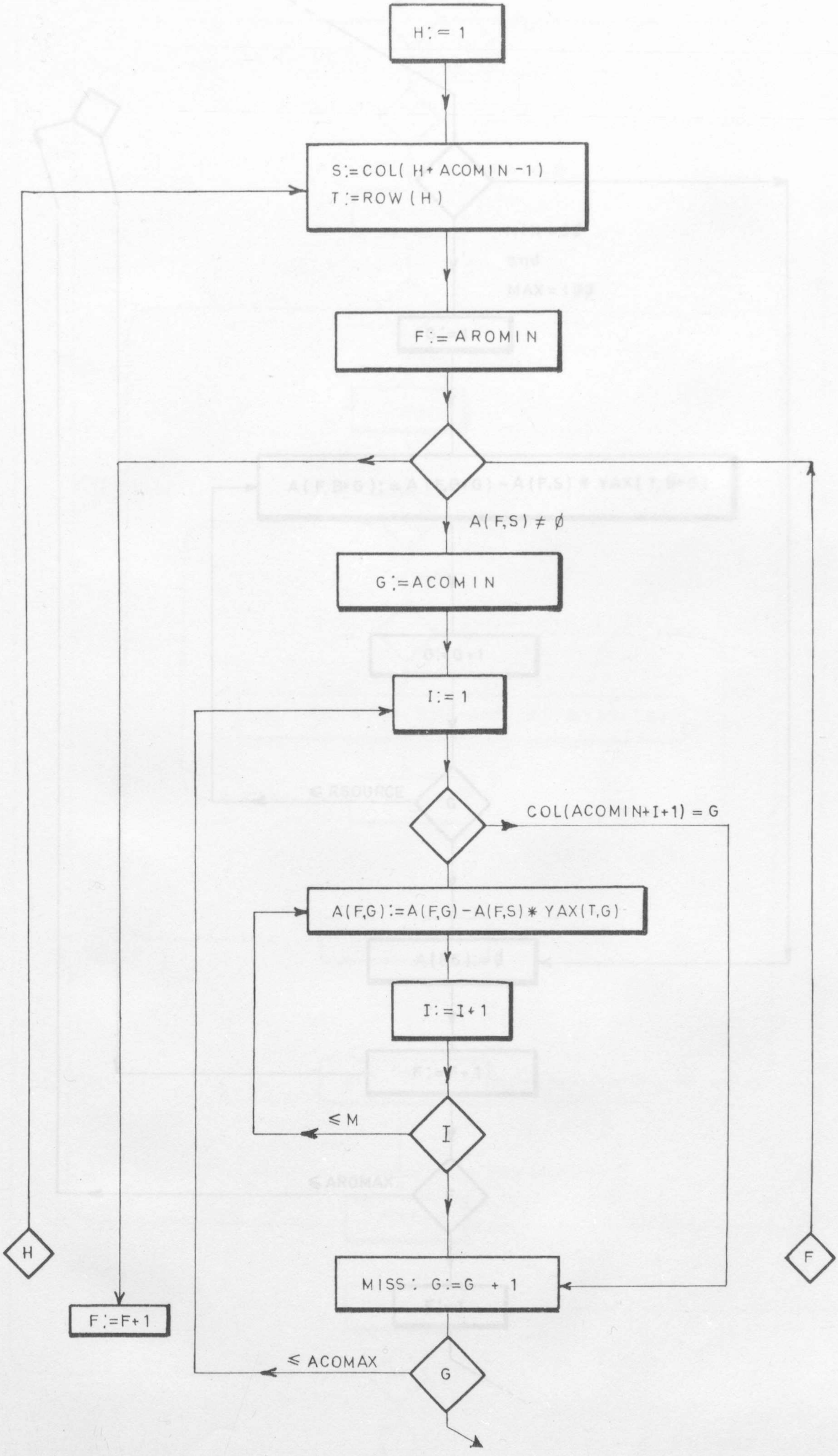
Procedure GFPIVOT sheet 5


MIN=5Ø
and
MAX=1ØØ
G:=1
YAX[ROW[M]B+G]:=
YAX[ROW[M]B+G/YAX[ROW[M],COL[M+LEAST-1]]
F:=M-1
H:=F+1
YAX[ROW[F],B+G]:=
YAX[ROW[F],B+G]-YAX[ROW[F],COL[H+LEAST 1]]*YAX[ROW[H],B+G]
H:=H+1
≤M
H
YAX[ROW[F],B+G]:=
YAX[ROW[F],B+G]/YAX[ROW[F],COL[F+LEAST-1]]
F:=F-1
≤1
F
G:=G+1
≤RSOURCE
G

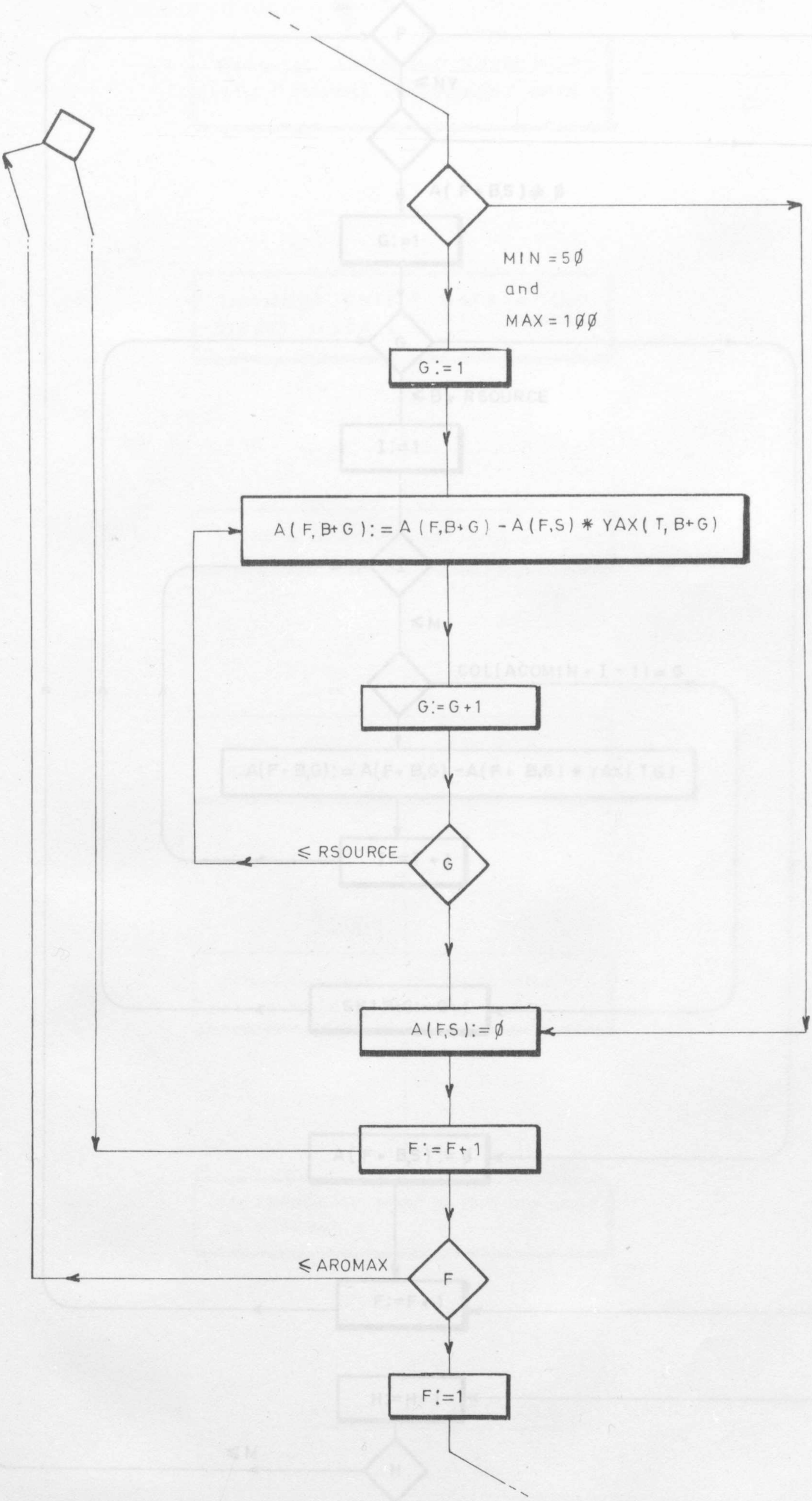
Procedure GFPIVOT sheet 6


H := 1
H th constraint variable of subsystem to be eliminated from MATRIX A
updates independent variable rows and columns of A
Is YAX a resistor subsystem MATRIX ?
YES
Updates source columns of A
NO
NO
Is A the main system MATRIX ?
YES
Updates independent variable columns of output state rows of A
Is YAX a resistor subsystem MATRIX ?
YES
Updates source columns of A
NO
NO
Have all constraint variables of subsystem been eliminated from A ?
YES

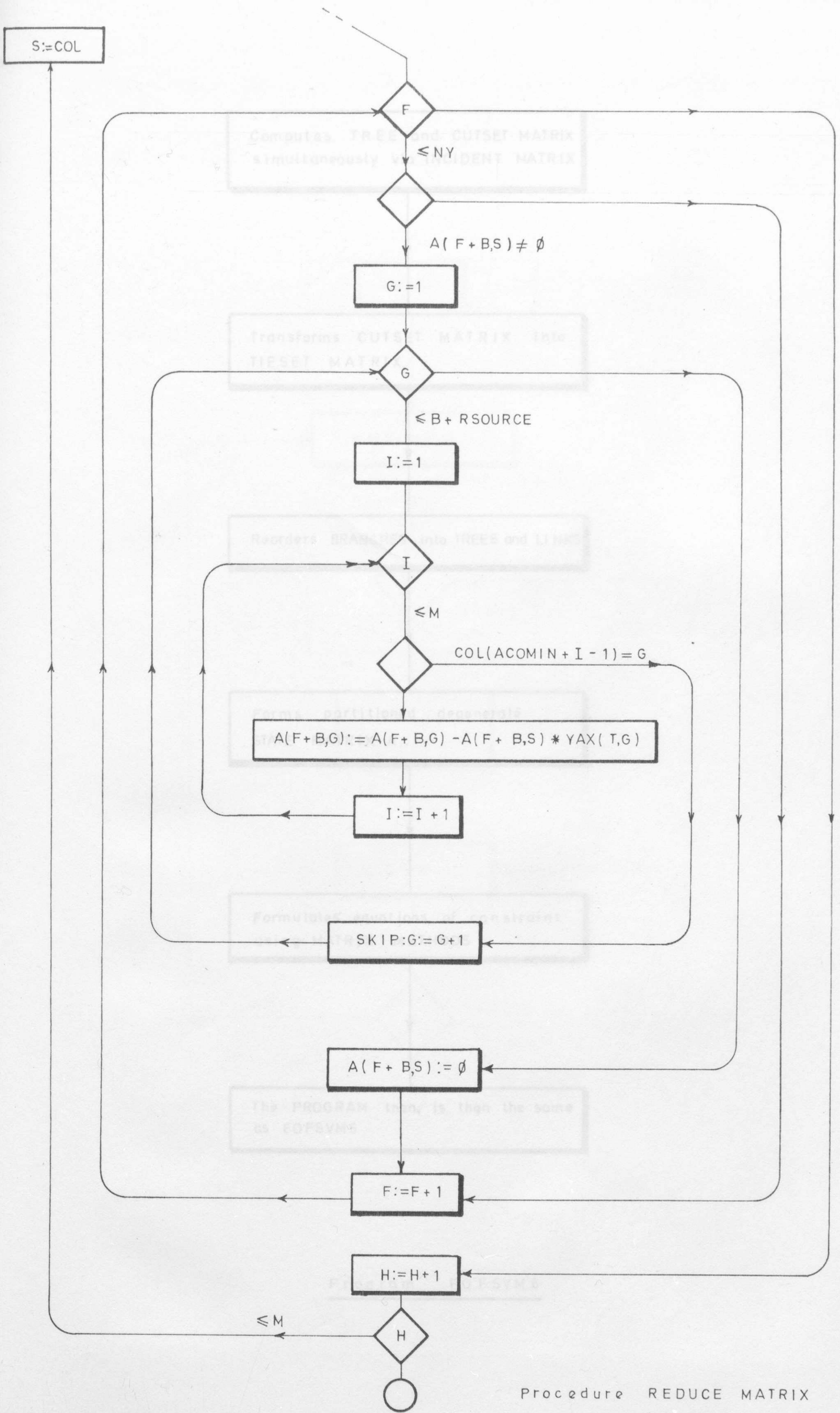
Procedure REDUCE MATRIX



Procedure REDUCE MATRIX (sheet 1)

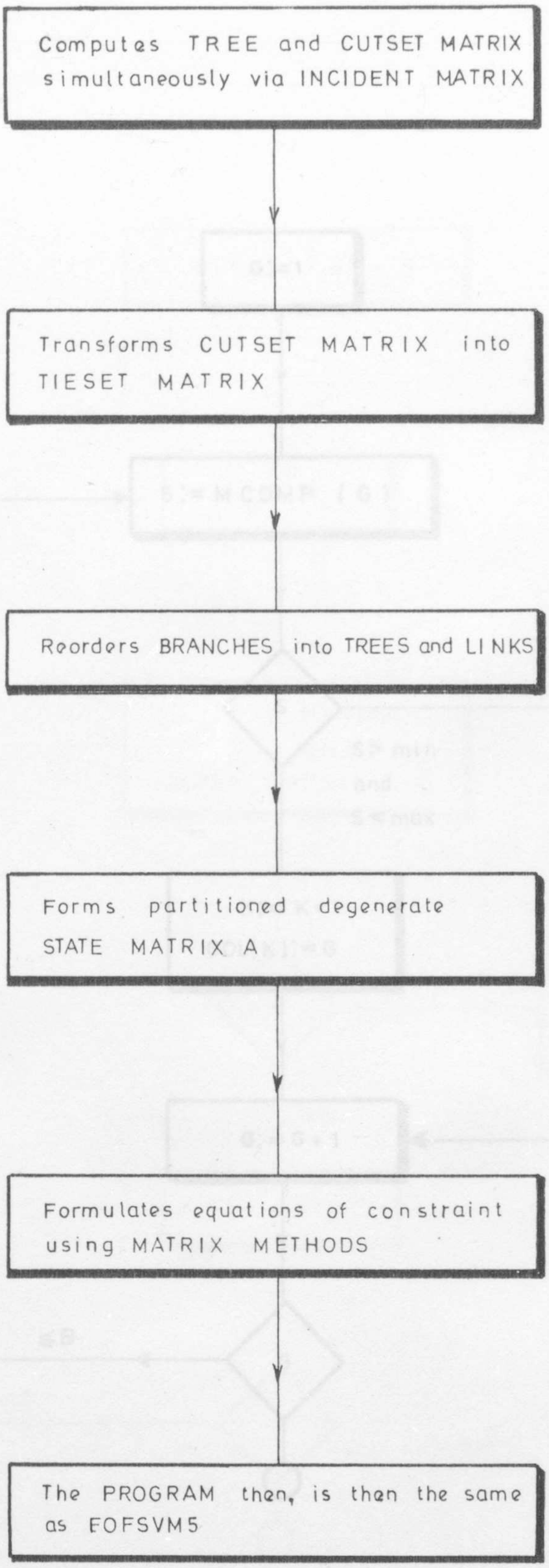


Procedure REDUCE MATRIX (sheet 2)

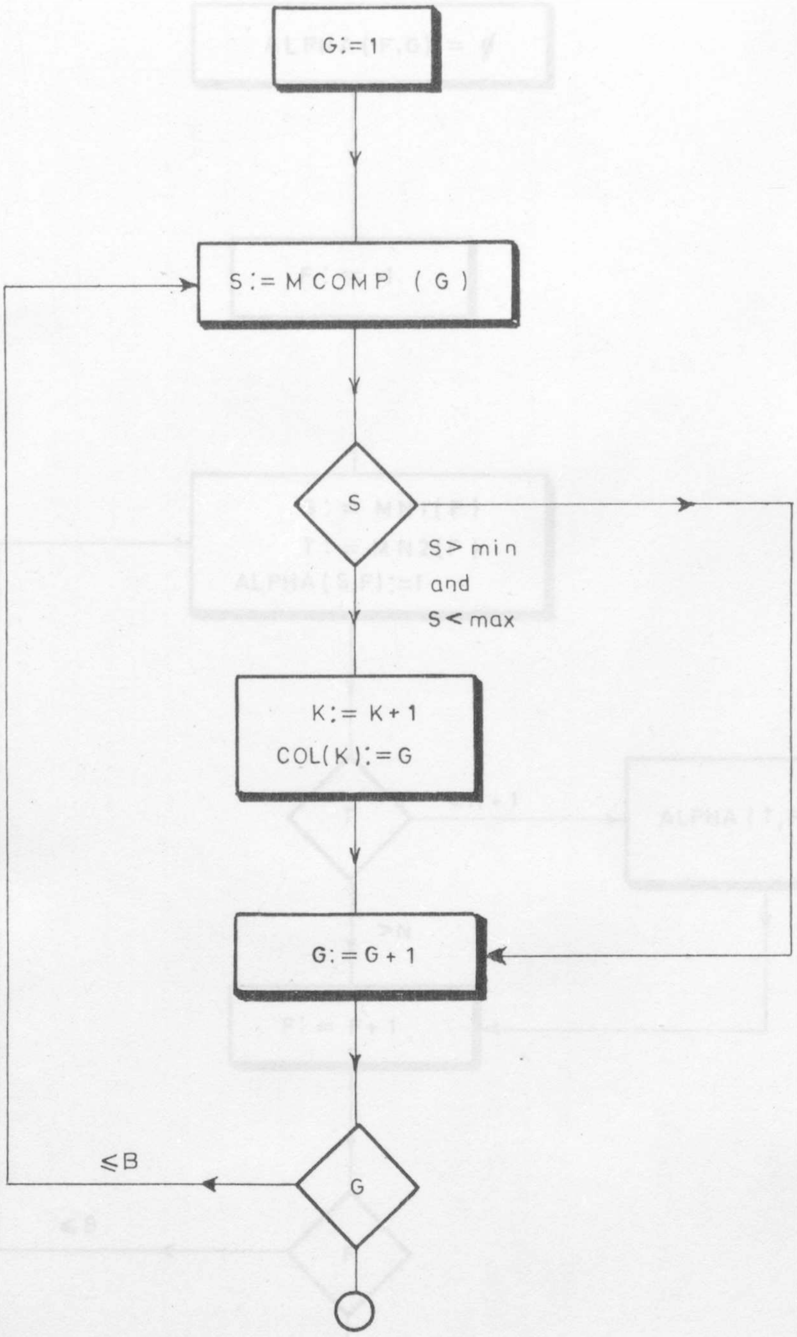


Procedure REDUCE MATRIX

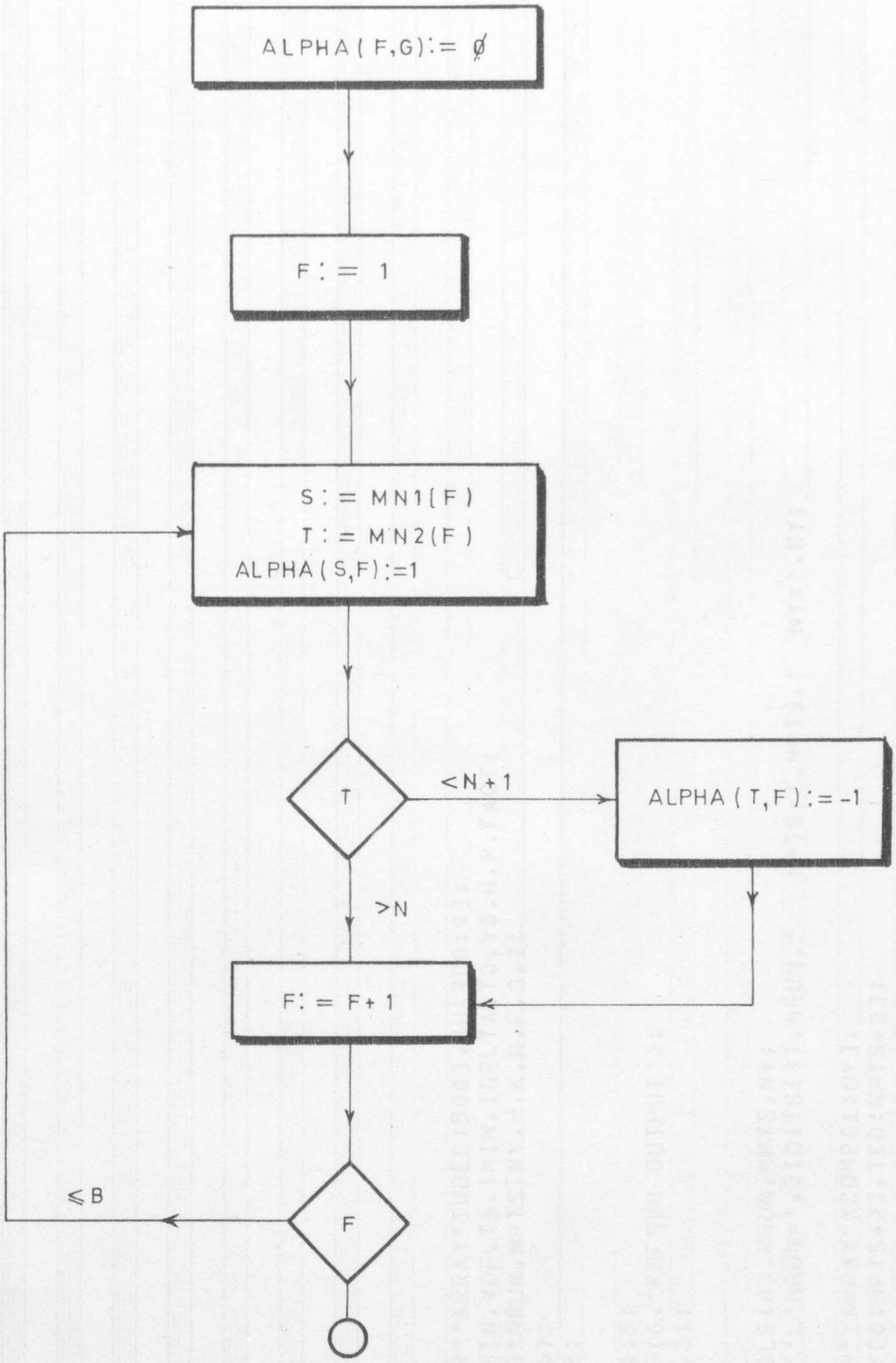
(sheet 3)


Computes TREE and CUTSET MATRIX simultaneously via INCIDENT MATRIX
Transforms CUTSET MATRIX into TIESET MATRIX
Reorders BRANCHES into TREES and LINKS
Forms partitioned degenerate STATE MATRIX A
Formulates equations of constraint using MATRIX METHODS
The PROGRAM then, is then the same as FOFSVM5

Program FOFSVM6



Procedure COSORT



Procedure INCIDENT MATRIX

```
&JOB;

&ALGOL;

&LIST;
      SVMPLT;
      "BEGIN"

&UNLIST;

&LIST;
      "INTEGER""ARRAY"TUB[0:500],CHAR[0:9];
      "REAL"VMIN,VDELTA,IMIN,IDELTA,T0,Y0,H,P,FACT;
      "INTEGER"NRUN,NPTS;NY,M,K,N,F,G,Z;
      READER(6);
      SAMELINE;
      M:=0;
       FACT:=1.5;
      MTSOURCE(6,‘FOFSVM OUTPUT’);
PLOTS(TUB,500,3);
NXTRUN:
      N:=0;
      "READ"FILE(6),NRUN,NPTS,NY;
      "PRINT"‘‘L’NRUN=’,DIGITS(4),NRUN,‘   NPTS=’,NPTS,‘  NY=’,NY;
      "BEGIN"
      "INTEGER""ARRAY"YCOMP[1:NY];
      "ARRAY"Y[0:NPTS+2],T[0:NPTS+2];
      FINDREC(6);
      "FOR"F:=1"STEP"1"UNTIL"NY"DO"
      "BEGIN"
      "READ"FILE(6),YCOMP[F];
      "PRINT"‘‘L’YCOMP=’,PREFIX(‘‘S3’’),DIGITS(4),YCOMP[F];
      "END";
      "FOR"G:=1"STEP"1"UNTIL"NY"DO"
      "BEGIN"
      "FOR"F:=0"STEP"1"UNTIL"NPTS+2"DO"
      "BEGIN"FINDREC(6);
      "READ"FILE(6),H;
      "IF"F=1"THEN"
      "BEGIN"
      "IF"H<₁₀-2"AND"H>₁₀-6"THEN"N:=3"ELSE"
      "IF"H<₁₀-5"AND"H>₁₀-9"THEN"N:=6"ELSE"
      "IF"H<₁₀-8"THEN"N:=9;
      T[0]:=10↑N*T[0];
      "END";
      T[F]:=H*10↑N;
      "IF"T[F]<0"THEN"
      "BEGIN""READ"FILE(6),NPTS;
      "PRINT"''L`NPTS=`,DIGITS(4),NPTS;
      "GOTO"NEXTG;
      "END";
      "FØR"K:=1"STEP"1"UNTIL"NY"DO"
      "BEGIN"
      "READ"FILE(6),H;
      "IF"G=K"THEN"
      "BEGIN"Y[F]:=H;
      "GOTO"NEXTF;
      "END""END";
NEXTF:"END";
NEXTG:
FACTOR(FACT);
      SCALE(T,10,NPTS,1);
      T0:=0;
      SCALE(Y,10,NPTS,1);
      Y0:=2;
      "IF"N=0"THEN"PUTSYM('  TIME  (SECS)   `,CHAR)"FLSE"
      "IF"N=3"THEN"PUTSYM('TIME (MILLISECS)`,CHAR)"ELSE"
      "IF"N=6"THEN"PUTSYM('TIME (MICROSECS)`,CHAR)"ELSE"
      PUTSYM('TIME  (NANOSECS)`,CHAR);
      AXIS(0,Y0,CHAR,-16,10.0,0.0,T[NPTS],T[NPTS+1]);
      "IF"YCOMP[K]=0"THEN"PUTSYM('CURRENT`,CHAR);
      "IF"YCOMP[K]=1"THEN"PUTSYM('VOLTAGE`,CHAR);
     AXIS(T0,2,CHAR,7,10.0,90.0,Y[NPTS],Y[NPTS+1]);
     PLOT(T0,Y0,-3);LINE(T,Y,NPTS,1,0,0);
      PLOT(-T0,-Y0,-3);
      PLOT(20,0,-3);
ENDBLOCK(6);
MTREWIND(6);
      MTSOURCE(6,'FOFSVM OUTPUT`);
      "FOR"K:=1"STEP"1"UNTIL"M+2"DO"
      FINDREC(6);
      "END";
      "END"OF BLOCK 2;
      "IF"NRUN>0"THEN"
      "BEGIN"
ENDBLOCK(6);
      MTREWIND(6);
      MTSOURCE(6,'FOFSVM OUTPUT`);
      M:=M+NPTS+3;
      "FOR"F:=1"STEP"1"UNTIL"M+1"DO"
      FINDREC(6);
      "GOTO"NXTRUN;
      "END";
      PLOT(-T0,-Y0,999);
      "END"OF SVMPLT;


&DIAG;


&RUN;
```

```
NRUN=     2   NPTS=  500   NY=      2
YCOMP=          1
YCOMP=          1
NPTS=  501
NPTS=  501
NRUN=     7   NPTS=  300   NY=      2
YCOMP=          1
YCOMP=          1
NPTS=  301
NPTS=  301
NRUN=     4   NPTS=  500   NY=      1
YCOMP=          0
NPTS=  261
NRUN=     6   NPTS=  500   NY=      1
YCOMP=          0
NPTS=   98
NRUN=     3   NPTS=  500   NY=      3
YCOMP=          1
YCOMP=          1
YCOMP=          1
NPTS=  242
NPTS=  242
NPTS=  242
NRUN=     5   NPTS=  500   NY=      2
YCOMP=          1
YCOMP=          1
NPTS=  104
NPTS=  104
NRUN=     6   NPTS=  300   NY=      2
YCOMP=          1
YCOMP=          1
NPTS=  139
NPTS=  139
NRUN=    -1   NPTS=  500   NY=      3
YCOMP=          1
YCOMP=          1
YCOMP=          1
NPTS=  366
NPTS=  366
```