CANINE

Computer Analysis of Networks via Inversion of Network Equations

by

Paul M. Russo

Memorandum No. ERL-M252

22 August 1968

CANINE

Computer Analysis of Networks via Inversion of Network Equations

by

Paul M. Russo

Memorandum No. ERL-M252

22 August 1968

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

ERRATA

Corrections to listing of CANINE: ERL-M252

SUBROUTINE PT:

Card C-246 should read

```
IF (KKTT, EO.1) go to 62
```

Card C-253 should be replaced by the following three cards:

```
I = NN - 1
DO 35 J = NN, NB
I = I + 1
```

The following card should be added right after C-257

```
62  I = I - 1
```

Comment on use of CANINE:

CANINE will not, in general, handle the following situations:

1. Compatible loops of voltage sources containing voltage controlled voltage sources.

2. Compatible cutsets of current sources containing current controlled current sources.

The above situations give rise to a singular $\hat{A}$ matrix, and no diagnostic is provided. For such problems, the use of CANDO[1] is recommended.

TABLE OF CONTENTS

ABSTRACT

This report describes the operation and use of CANINE (Computer Analysis of Networks via Inversion of Network Equations), a time-domain analysis program, for linear time-invariant networks. The networks may contain dependent and independent sources of all types,capacitances, resistances and inductances.

The network analysis problem is to obtain the branch currents and voltages by solving a set of simultaneous differentio-algebraic equations derived from the branch relations and Kirchhoff's current and voltage laws.

In the program CANINE, tree voltages and link currents form a basis set of variables, denoted by $\underset{\sim}{x}$. This formulation yields the automatic satisfaction of Kirchhoff's laws. A numerical integration formula reduces the system to a set of simultaneous algebraic equations of the form

$$\underset{\sim}{A}\,\underset{\sim}{x} = \underset{\sim}{k},$$

where $\underset{\sim}{A}$ is called the "network equation matrix" and $\underset{\sim}{k}$ is a vector, which is constant at any time point, and which will be defined later.

The solution $\underset{\sim}{x}$ is obtained by inverting the matrix A:

$$\underset{\sim}{x} = \underset{\sim}{A}^{-1}\,\underset{\sim}{k}\,.$$

The tree-picking and internal current scaling algorithms are such that large value spread and large time constant spread problems can be handled reasonably effectively and efficiently.

The program CANINE is written in Fortran IV for the CDC 6400 computer operating under the (CAL) Scope 3.1 system.

ACKNOWLEDGEMENTS

I wish to acknowledge the constant encouragement and motivating influence of Professor R. A. Rohrer, without whose guidance the writing of this report would have never taken place.

I also wish to acknowledge the financial support of the Air Force Office of Scientific Research, under Grant AF-AFOSR-1219-67, and the Computer Center of the University of California at Berkeley, which has been most gracious in its granting of subsidized computer time.

INTRODUCTION

Since the program CANINE evolved from the program CANDO[1], many of the subroutines making up CANINE are identical to those of program CANDO.

The following subroutines are common to both programs and are described in (1);

Subroutine READIN

Subroutine PT

Subroutine FCSM

Subroutine KONST

Subroutine CALCAL

Subroutine INCREM

Subroutine PERIOD

Subroutine READOUT

Subroutine GRAPH

Subroutine ALLOUT

Knowledge of the operation of the above subroutines is not necessary for the effective use of program CANINE.

It should be noted, at this point, that programs CANINE and CANDO are complementary in the sense that each is better suited to a particular type of network analysis. If an exact analysis is desired, CANINE has been found to be the superior (in the sense of computer time) whereas, if only an approximate analysis is desired (as in the initial stages of automated network design), CANDO, due to its iterative formulation, is the more desirable program.

The choice of either program over the other must depend on the type of analysis that the user finds most desirable, for his particular application.

Section I describes the formulation of the matrix $\underset{\sim}{A}$ and the vector $\underset{\sim}{k}$. Section II describes the main program and subroutines associated with CANINE which have not been employed in the program CANDO. The Appendices describe the use of CANINE in solving network problems, and the way dependent sources must be modeled.

SECTION I: NETWORK ANALYSIS

Theory:

The solution of a network consists of finding the branch current and voltage vectors $\underset{\sim}{i}_b$ and $\underset{\sim}{v}_b$ respectively. The solution is obtained by solving the following simultaneous set of equations

$\underset{\sim}{f}_b(\underset{\sim}{i}_b, \underset{\sim}{v}_b) = \underset{\sim}{0}$ branch relations

$\underset{\sim}{\mathcal{Q}}\,\underset{\sim}{i}_b = \underset{\sim}{0}$ Kirchhoff's current law

$\underset{\sim}{B}'\,\underset{\sim}{v}_b = \underset{\sim}{0}$ Kirchhoff's voltage law

where $\underset{\sim}{\mathcal{Q}}$ is the fundamental cutset matrix, and $\underset{\sim}{B}$ is the fundamental loop matrix based on an appropriately selected tree. We may renumber the NB branches of our network in such a way that the 1st NN-1 (where NN is the number of nodes in our network) branches form the tree.

With the above numbering scheme, $\mathcal{Q}$ and B may be partitioned as follows:

$$\underset{\sim}{\mathcal{Q}} = \left[\underset{\sim}{I} \ : \ \underset{\sim}{F}\right] \quad \text{and } \underset{\sim}{B} = \left[-\underset{\sim}{F}' \ : \ \underset{\sim}{I}\right]$$

and hence we have, from Kirchhoff's laws

$$\underset{\sim}{i}_t = -\underset{\sim}{F}\, i_\ell$$

$$\underset{\sim}{v}_\ell = \underset{\sim}{F}'\, \underset{\sim}{v}_t,$$

where the subscripts t and ℓ refer to tree branches and links respectively. Our tree picking algorithms are discussed in subroutine PT[1], suffice it to say that in both our tree-picking schemes, independent voltage sources must be tree branches and independent current sources must be links. Upon using a numerical integration formula (see subroutine MATSET), our set of equations reduces to

$$\underset{\sim}{\ell}(\underset{\sim}{v}_t, \underset{\sim}{i}_\ell) = \underset{\sim}{0}.$$

This set of linear equations is solved by reducing it to the form $\underset{\sim}{A}\,\underset{\sim}{x} = \underset{\sim}{k}$ (see subroutine MATSET) where

$$\underset{\sim}{x} = \begin{bmatrix} \underset{\sim}{v}_t \\ \hline \underset{\sim}{i}_\ell \end{bmatrix}$$

and inverting $\underset{\sim}{A}$ to yield

$$\underset{\sim}{x} = \underset{\sim}{A}^{-1}\underset{\sim}{k}.$$

The solution obtained by CANINE is exact and is limited only by roundoff error in the computer and the accuracy of the integration formula.

<u>Solving the Initial Condition Problem:</u>

Given the initial capacitance voltages and inductance currents, we wish to compute $\underset{\sim}{v}_1$ and $\underset{\sim}{i}_1$, the branch voltages and currents, at the initial time point. This reduces to the problem of solving a coupled system of algebraic equations.

A <u>proper</u> tree is selected according to the algorithm described in subroutine PT[1]. This algorithm maximizes the number of capacitances in tree branches and the number of inductances in links.

If there are no capacitance or capacitance-voltage source loops, or inductance or inductance-current source cutsets, all capacitances will be tree branches, and all inductances will be links.

Kirchhoff's current and voltage laws become automatically satisfied by selecting the tree voltages and link currents to be a basis set of variables, with the following being true

$$\underset{\sim}{i}_{t_1} = -\underset{\sim}{F}\,\underset{\sim}{i}_{\ell_1}$$

and

$$\underset{\sim}{v}_{\ell_1} = \underset{\sim}{F}'\,\underset{\sim}{v}_{t_1},$$

where the subscripts ℓ and t refer to links and tree branches respectively, and where $\underset{\sim}{\mathcal{Q}} = \left[\underset{\sim}{I} : \underset{\sim}{F}\right]$ is the fundamental cutset matrix based on the proper tree.

Tree capacitances are treated as independent voltage sources with their voltage being set to the initial condition. Similarly, link inductances are treated as independent current sources with their current being set to the initial condition. Tree inductance voltages and link capacitance currents are set equal to zero, and their inital conditions are lost. Hence, incompatible initial conditions (e.g., incompatible capacitance voltages around a capacitance-voltage source loop or incompatible inductor currents at an inductor-current source cutset) are forced to be compatible, with the tree inductances and link capacitances loosing their specified initial conditions.

Except for the fact that reactive elements are treated as indpendent sources, the solution of the initial condition problem is analogous to that of the "general time point problem," and will not be detailed here. Suffice it to say that our system of equations is reduced to the form

$$\underset{\sim}{A}\,\underset{\sim}{x} = \underset{\sim}{k}$$

and the solution is obtained, via the inversion of $\underset{\sim}{A}$, from

$$\underset{\sim}{x} = \underset{\sim}{A}^{-1}\,\underset{\sim}{k},$$

where

$$\underset{\sim}{x} = \begin{bmatrix} \underset{\sim}{v}_{t_1} \\ \hdashline \underset{\sim}{i}_{t_1} \end{bmatrix} .$$

Solving the General Time Point Problem:

Knowing $\underset{\sim}{i}_1$ and $\underset{\sim}{v}_1$, the branch currents and voltages at the previous time point, we wish to compute $\underset{\sim}{i}$ and $\underset{\sim}{v}$, the branch currents and voltages at the succeeding time point. The problem reduces to solving the coupled system of differential and algebraic equations, consisting of the branch relations, and Kirchhoff's current and voltage laws.

A tree is selected according to the optimal tree algorithm of subroutine PT[1], rendering the system well behaved, and hence minimizing computational errors in our matrix inversion scheme. The network is renumbered such that the first NN-1 branches form the tree, and the subsequent branches are links. Kirchhoff's current and voltage laws become automatically satisfied by selecting the tree voltages and link currents as a basis set of variables. Note that this approach requires that independent voltage sources be tree branches and that independent current sources be links.

The tree currents $\underset{\sim}{i}_t$ are given, in terms of the link currents $\underset{\sim}{i}_\ell$ by

$$\underset{\sim}{i}_t = -\underset{\sim}{F}\,\underset{\sim}{i}_\ell ,$$

where $\underset{\sim}{\mathcal{Q}} = \left[\underset{\sim}{I} \;\vdots\; \underset{\sim}{F}\right]$ is the fundamental cutset matrix associated with our <u>optimal</u> tree. Similarly,

$$\underset{\sim}{v}_\ell = \underset{\sim}{F}'\,\underset{\sim}{v}_t ,$$

where $\underset{\sim}{F}'$ is the transpose of $\underset{\sim}{F}$.

The problem thus reduces to solving the set of equations

$$\underset{\sim}{f}\,(\underset{\sim}{v}_t, \dot{\underset{\sim}{v}}_t, \underset{\sim}{i}_t, \dot{\underset{\sim}{i}}_t) = \underset{\sim}{0}$$

Integrating the above set of equations, via a numerical integration formula, reduces the problem to

$$L(\underset{\sim}{v}_t , \underset{\sim}{i}_\ell) = \underset{\sim}{0} ,$$

where the L_i are the general time point branch relations, in terms of our basis set of variables. The above set of equations is reduced to the form

$$\underset{\sim}{A}\,\underset{\sim}{x} = \underset{\sim}{k} ,$$

and the solution is obtained, explicitely, from

$$\underset{\sim}{x} = \begin{bmatrix} \underset{\sim}{v}_t \\ \hdashline \underset{\sim}{i}_\ell \end{bmatrix} = \underset{\sim}{A}^{-1} \underset{\sim}{k} .$$

Construction of $\underset{\sim}{A}$ and $\underset{\sim}{k}$:

We wish to solve a set of branch relations of the form

$$\underset{\sim}{f}(\underset{\sim}{v}_b, \underset{\sim}{i}_b) = \underset{\sim}{0} \tag{1}$$

while simultaneously satisfying Kirchhoff's voltage and current laws. Upon the selection of a tree, and appropriate renumbering, Kirchhoff's voltage and current laws may be expressed as

$$\left.\begin{aligned} \underset{\sim}{v}_\ell &= \underset{\sim}{F}' \underset{\sim}{v}_t \\ \underset{\sim}{i}_t &= - \underset{\sim}{F} \underset{\sim}{i}_\ell \end{aligned}\right\} \tag{2}$$

where the subscripts t and ℓ refer to tree branches and links, respectively, and

$$\underset{\sim}{\mathcal{Q}} = \left[\underset{\sim}{I} \; : \; \underset{\sim}{F} \right]$$

is the fundamental cutset matrix based on our topological tree.

Substituting (2) into (1), we obtain

$$\underset{\sim}{L}(\underset{\sim}{v}_t, \underset{\sim}{i}_\ell) = \underset{\sim}{0} \tag{3}$$

We wish to reduce (3) to the form

$$\underset{\sim}{A} \, \underset{\sim}{x} = \underset{\sim}{k} \tag{4}$$

where

$$\underset{\sim}{x} = \begin{bmatrix} \underset{\sim}{v}_t \\ ---- \\ \underset{\sim}{i}_\ell \end{bmatrix} \tag{5}$$

and hence the solution is given by

$$\underset{\sim}{x} = \underset{\sim}{A}^{-1} \underset{\sim}{k} \tag{6}$$

Note that $\underset{\sim}{A}$ is an NB × NB nonsingular matrix, where NB is the number of branches in the network.

Upon the inclusion of a numerical integration formula, the ith general branch relation can be written in the form

$$-\alpha_i \cdot i_i + \beta_i \cdot v_i - C_i \cdot i_m - \ell_i \cdot v_m = k_i \tag{7}$$

The C_i and ℓ_i terms are nonzero only for dependent sources, and m designates the controlling branch.

Tree Branches:

NN designates the number of nodes, P = NN - 1, and $const_i$ designates the branch constant as computed in subroutine KONST (1). H denotes the integration step size.

Independent Voltage Sources:

branch relation

v_i = source value

Thus $\alpha_i = C_i = \ell_i = 0$

$\beta_i = 1$

k_i = source value

so $A_{ij} = \delta_{ij}$ (Kronecker delta)

$= 1 \quad i = j$

$= 0 \quad i \neq j$

Dependent Voltage Source:

a) voltage controlled (controlling voltage is a link voltage)).

branch relation:

$$v_i - \text{const}_i \cdot v_m = 0$$

Thus $\alpha_i = C_i = k_i = 0$

$\beta_i = 1$

$\ell_i = \text{const}_i$

since v_m is a link voltage, we must express it in terms of tree voltages, i. e.,

$$v_m = \sum_{K} F'_{m-P, K} \cdot v_K$$

over tree branches

so $A_{ij} = 0 \quad j \geq NN$

$A_{ij} = \delta_{ij} - F_{j, m-P} \cdot \text{const}_i \quad j < NN$

b) current controlled (controlling current is a tree current).

branch relation:

$$v_i - \text{const}_i \, i_m = 0$$

thus $\alpha_i = \ell_i = k_i = 0$

$C_i = \text{const}_i$

$\beta_i = 1$

since i_m is a tree current, we must express it in terms of link currents, i.e.,

$$i_m = - \sum_{\substack{K \\ \text{over links}}} F_{mK} \cdot i_K$$

so, $A_{ij} = \delta_{ij} \quad j < NN$

$A_{ij} = \text{const}_i \cdot F_{m,j-P} \quad j \geq NN$

Tree Capacitances:

a) initial time point problem

branch relation:

v_i = initial condition

Thus $\alpha_i = C_i = \ell_i = 0$

$\beta_i = 1$

k_i = initial condition

so $A_{ij} = \delta_{ij}$

b) general time point problem

Let 1 designate the previous time point and 2 designate the present time point. A trapezoidal numerical integration formula is employed.

branch relation:

$$v_{i2} - \text{const}_i \; i_{i2} = \text{const}_i \cdot i_{i1} + v_{i1}$$

Thus $k_i = \text{const}_i \cdot i_{i1} + v_{i1}$

$$\ell_i = C_i = 0$$

$$\beta_i = 1$$

$$\alpha_i = \text{const}_i$$

Now i_{i2} must be expressed in terms of link currents, i. e.,

$$i_{i2} = - \sum_{\substack{K \\ \text{over links}}} F_{iK} \cdot i_{2_K}$$

so $A_{ij} = \delta_{ij} \qquad j < NN$

$$A_{ij} = \text{const}_i \cdot F_{i,j-P} \qquad j \geq NN$$

Tree Resistance:

branch relation:

$$v_i - \text{const}_i \cdot i_i = 0$$

Thus $\alpha_i = \text{const}_i$

$$\beta_i = 1$$

$$C_i = \ell_i = k_i = 0$$

We must express i_i in terms of link currents, i. e.,

$$i_i = -\sum_{\substack{K \\ \text{over links}}} F_{iK} \cdot i_K$$

so, $A_{ij} = \delta_{ij} \quad j < NN$

$A_{ij} = \text{const}_i \cdot F_{i,j-P} \quad j \geq NN$

Tree Inductances:

a) initial time point problem

Due to our selection of a proper tree, the initial voltage across a link inductance is taken to be zero.

branch relation:

$v_i = 0$

Thus $\alpha_i = C_i = \ell_i = k_i = 0$

so $A_{ij} = \delta_{ij}$

b) general time point problem

Let 1 designate the previous time point and 2 designate the present time point. A trapezoidal numerical integration formula is employed.

branch relation:

$v_{i2} - \text{const}_i \cdot i_{i2} = - v_{i1} - \text{const}_i \cdot i_{i1}$

thus $\alpha_i = \text{const}_i$

$\beta_i = 1$

$C_i = \ell_i = 0$

$$k_i = - v_{i1} - \text{const}_i \cdot i_{i1}$$

We must express i_{i2} in terms of link currents, i.e.,

$$i_{i2} = - \sum_{\substack{K \\ \text{over links}}} F_{iK} \cdot i_{2_K}$$

so $A_{ij} = \delta_{ij} \quad j < NN$

$A_{ij} = \text{const}_i \cdot F_{i, j-P} \quad j \geq NN$

<u>Dependent Current Sources:</u>

a) voltage controlled (controlling voltage is a link voltage).

branch relation:

$$i_i - \text{const}_i \cdot v_m = 0$$

thus $\alpha_i = -1$

$\beta_i = C_i = k_i = 0$

$\ell_i = \text{const}_i$

We must express v_m in terms of tree voltages, and i_i in terms of link currents, i.e.,

$$v_m = \sum_{\substack{K \\ \text{over tree} \\ \text{branches}}} F'_{m-P, K} \cdot v_K$$

$$i_i = - \sum_{\substack{K \\ \text{over links}}} F_{iK} \cdot i_K$$

so $A_{ij} = -F_{j,m-P} \cdot const_i$ $\quad j < NN$

$A_{ij} = -F_{i,j-P}$ $\quad j \geq NN$

b) current controlled (controlling current is a tree current).

branch relation:

$$i_i - const_i \cdot i_m = 0$$

thus $\alpha_i = -1$

$$\beta_i = \ell_i = k_i = 0$$

$$C_i = const_i$$

Both i_i and i_m are tree currents and must be expressed in terms of link currents, i.e.,

$$i_i = -\sum_{\substack{K \\ \text{over links}}} F_{iK} \cdot i_K \,, \qquad i_m = -\sum_{\substack{K \\ \text{over links}}} F_{mK} \cdot i_K$$

so $A_{ij} = 0$ $\quad j < NN$

$A_{ij} = -F_{i,j-P} + const_i \cdot F_{m,j-P}$ $\quad j \geq NN$

Link Branches:

Independent Current Sources:

branch relation:

$$i_i = \text{source value}$$

thus $\alpha_i = -1$

$\beta_i = C_i = \ell_i = 0$

k_i = source value

so, $A_{ij} = \delta_{ij}$

Dependent Current Sources:

a) voltage controlled (controlling branch is a link)

branch relation:

$$i_i - \text{const}_i \cdot v_m = 0$$

thus $\alpha_i = -1$

$\beta_i = C_i = k_i = 0$

$\ell_i = \text{const}_i$

We must express v_m in terms of tree voltages, i. e.,

$$v_m = \sum_{\substack{K \\ \text{tree branches}}} F'_{m-P,K} \cdot v_K$$

so, $A_{ij} = - F_{j,m-K} \cdot \text{const}_i$ $\qquad j < NN$

$A_{ij} = \delta_{ij}$ $\qquad j \geq NN$

b) current controlled (controlling branch is a tree branch).

branch relation:

$$i_i - \text{const}_i \cdot i_m = 0$$

thus $\alpha_i = -1$

$$\beta_i = \ell_i = k_i = 0$$

$$C_i = \text{const}_i$$

We must express i_m in terms of link currents, i.e.,

$$i_m = -\sum_{\substack{K \\ \text{over links}}} F_{mK} \cdot i_K$$

so, $A_{ij} = 0 \quad j < NN$

$$A_{ij} = \delta_{ij} + F_{m,j-P} \cdot \text{const}_i \quad j \geq NN$$

<u>Link Capacitances:</u>

a) initial time point problem

branch relation:

$$i_i = 0$$

thus $\alpha_i = -1$

$$\beta_i = C_i = \ell_i = k_i = 0$$

so, $A_{ij} = \delta_{ij}$

b) general time point problem

Let the subscripts 1 and 2 designate the previous and present time points respectively. A trapezoidal numerical integration formula is employed.

branch relation:

$$i_{i2} - \text{const}_i \cdot v_{i2} = -\text{const}_i \cdot v_{i1} - i_{i1}$$

thus $\alpha_i = -1$

$$\beta_i = -\text{const}_i$$

$$C_i = \ell_i = 0$$

$$k_i = -\text{const}_i \cdot v_{i1} - i_{i1}$$

We must express v_{i2} in terms of tree voltages, i. e.,

$$v_{i2} = \sum_{\substack{K \\ \text{tree branches}}} F'_{i-P} \cdot v_{2_K}$$

so, $A_{ij} = -F_{j,i-P} \cdot \text{const}_i \qquad j < NN$

$A_{ij} = \delta_{ij} \qquad j \geq NN$

Link Resistance:

branch relation:

$$i_i - \text{const}_i v_i = 0$$

thus $\alpha_i = -1$

$$\beta_i = -\text{const}_i$$

$$C_i = \ell_i = k_i = 0$$

We must express v_i in terms of tree voltages, i. e.,

$$v_i = \sum_{\substack{K \\ \text{tree branches}}} F'_{i-P,K} \cdot v_K$$

so, $A_{ij} = - F_{j,i-\ell} \cdot \text{const}_i$ $\qquad j < NN$

$A_{ij} = \delta_{ij}$ $\qquad j \geq NN$

Link Inductance:

a) initial time point problem.

branch relation:

i_i = initial condition

thus $\alpha_i = - 1$

$\beta_i = C_i = \ell_i = 0$

k_i = initial condition

so, $A_{ij} = \delta_{ij}$

b) general time point problem.

The subscripts 1 and 2 designate the previous and present time points respectively. A trapezoidal numerical integration formula is used.

branch relation:

$$i_{i2} - \text{const}_i \cdot v_{i2} = i_{i1} + \text{const}_i \cdot v_{i1}$$

thus $\alpha_i = - 1$

$\beta_i = -\text{const}_i$

$$C_i = \ell_i = 0$$

$$k_i = i_{i1} + \text{const}_i \cdot v_{i1}$$

We must express v_{i2} in terms of tree voltages, i.e.,

$$v_{i2} = \sum_{\substack{K \\ \text{tree branches}}} F'_{i-P,K} \cdot v_{2_K}$$

so, $A_{ij} = -F_{j,i-P} \cdot \text{const}_i \quad j < NN$

$A_{ij} = \delta_{ij} \quad j \geq NN$

<u>Dependent Voltage Sources:</u>

a) voltage controlled (controlling branch is a link).

branch relation:

$$v_i - \text{const}_i \cdot v_m = 0$$

thus $\alpha_i = C_i = k_i = 0$

$$\beta_i = 1$$

$$\ell_i = \text{const}_i$$

both v_i and v_m must be expressed in terms of tree voltages, i.e.,

$$v_i = \sum_{\substack{K \\ \text{tree branches}}} F'_{i-P,K} \cdot v_K$$

$$v_m = \sum_{\substack{K \\ \text{tree branches}}} F'_{m-P,K} \cdot v_K$$

so, $A_{ij} = F_{j,i-P} - F_{j,m-P} \cdot \text{const}_i \qquad j < NN$

$A_{ij} = 0.0 \qquad j \geq NN$

b) current controlled (controlling branch is a tree branch).

branch relation:

$$v_i - \text{const}_i \cdot i_m = 0$$

thus $\alpha_i = \ell_i = K_i = 0$

$\beta_i = 1$

$C_i = \text{const}_i$

We must express v_i in terms of tree voltages, and i_m in terms of link currents, i. e.,

$$v_i = \sum_{\substack{K \\ \text{tree branches}}} F'_{i-P,K} \cdot v_K$$

$$i_m = \sum_{\substack{K \\ \text{link branches}}} F_{m-P,K} \cdot v_K$$

so, $A_{ij} = F_{j,i-P} \qquad j < NN$

$A_{ij} = F_{m-P,j-P} \cdot \text{const}_i \qquad j \geq NN$

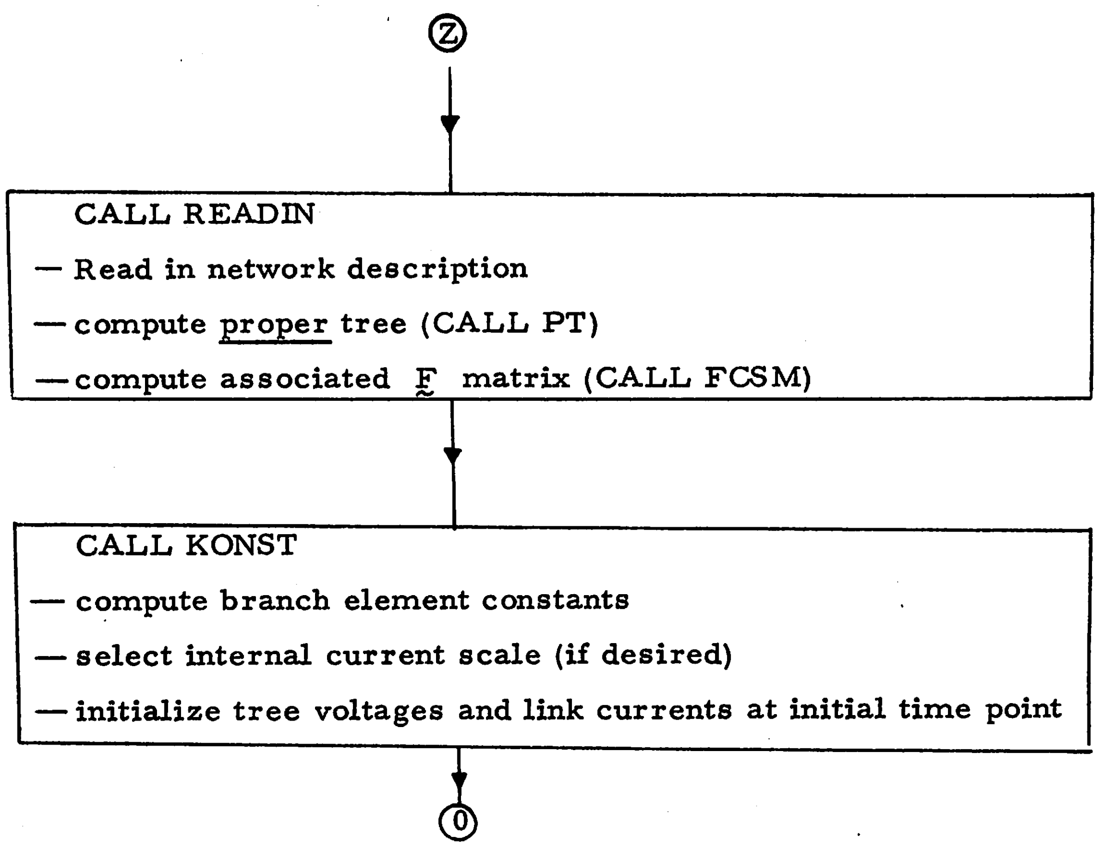
SECTION II

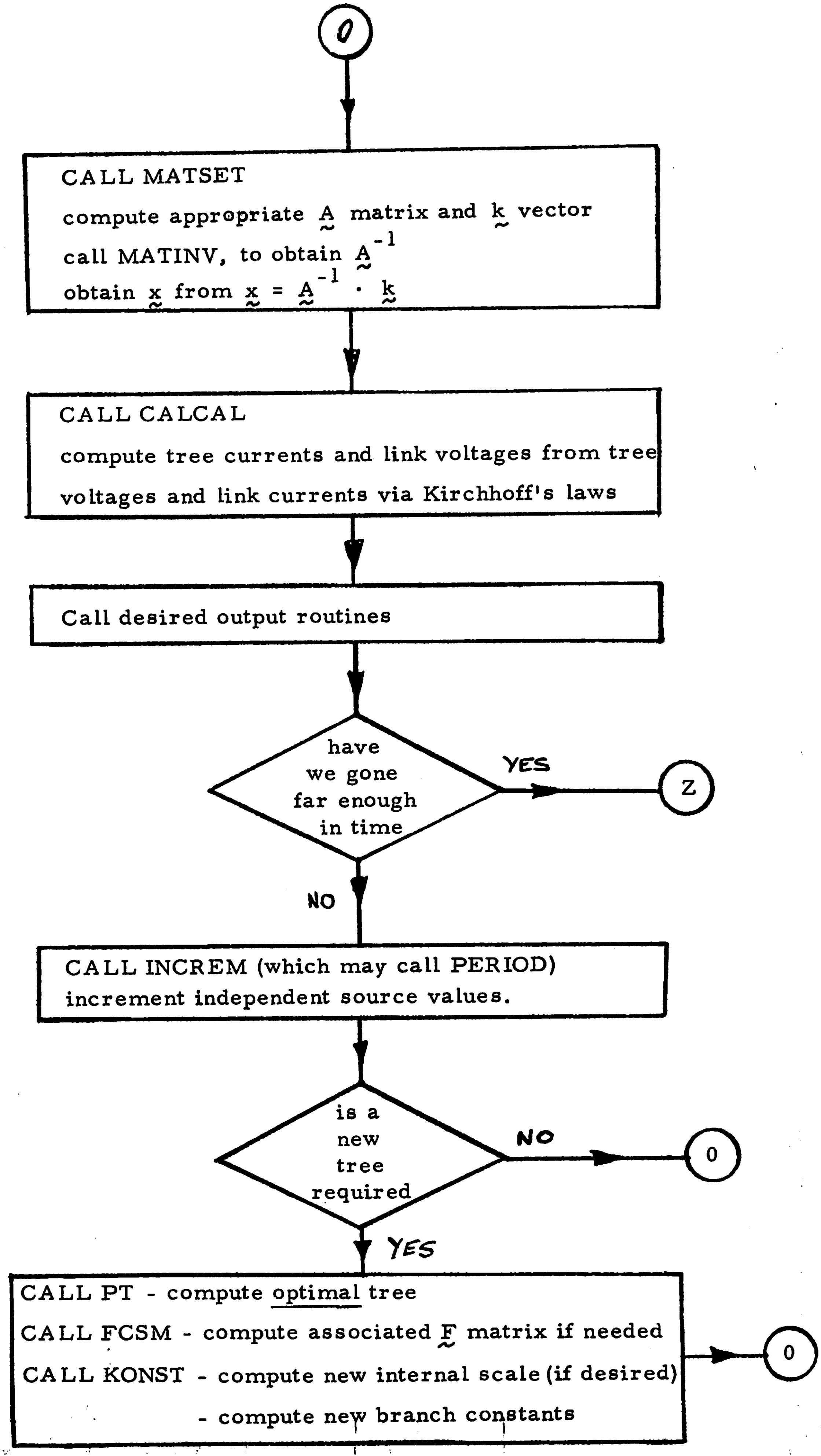
Program CANINE

MAIN program:

The MAIN section of program CANINE ensures that the desired subroutines are called in the correct sequence. It solves both the initial condition and general time point problems, in the way discussed in Section I. It also controls the calls to the output routines.

BLOCK DIAGRAM - MAIN PROGRAM




0
CALL MATSET
compute appropriate A matrix and k vector
call MATINV, to obtain A⁻¹
obtain x from x = A⁻¹ · k
CALL CALCAL
compute tree currents and link voltages from tree
voltages and link currents via Kirchhoff's laws
Call desired output routines
have we gone far enough in time
YES
Z
NO
CALL INCREM (which may call PERIOD)
increment independent source values.
is a new tree required
NO
0
YES
CALL PT - compute optimal tree
CALL FCSM - compute associated F matrix if needed
CALL KONST - compute new internal scale (if desired)
- compute new branch constants
0

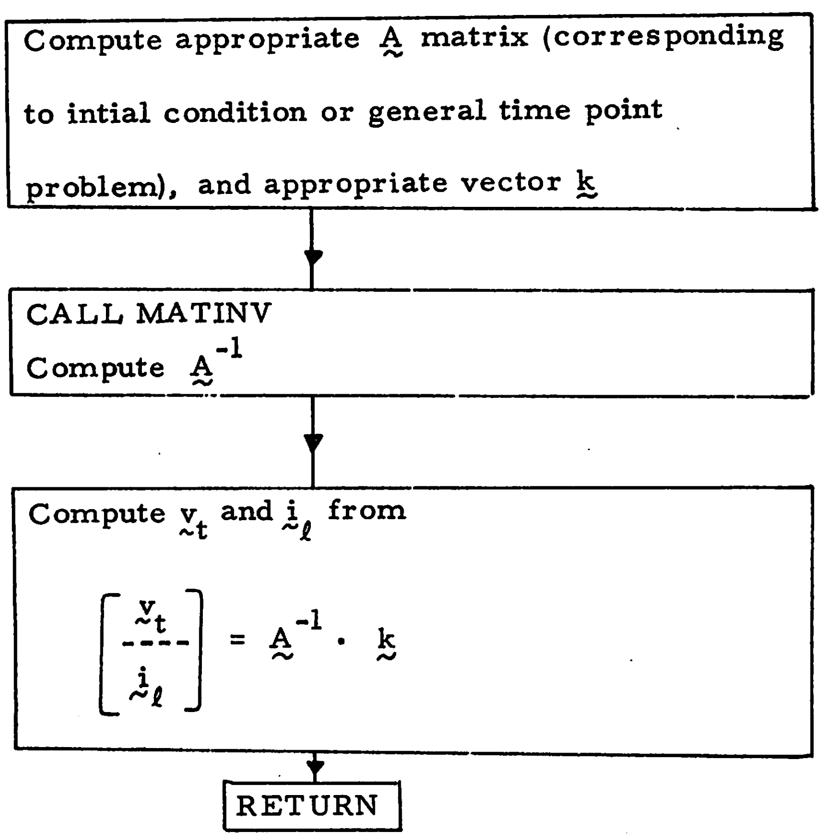
SUBROUTINE MATSET

This subroutine computes the $\underset{\sim}{A}$ matrix and $\underset{\sim}{k}$ vector for both the initial condition problem and for the general time point problem. MATSET then calls MATINV which computes $\underset{\sim}{A}^{-1}$. The solution is then computed from

$$\begin{bmatrix} \underset{\sim}{v}_t \\ \text{----} \\ \underset{\sim}{i}_\ell \end{bmatrix} = \underset{\sim}{A}^{-1} \cdot \underset{\sim}{k}$$

The general operation of this subroutine may be seen below.

BLOCK DIAGRAM - SUBROUTINE MATSET



SUBROUTINE MATINV

This is a standard IBM matrix inversion routine which obtains $\underset{\sim}{A}^{-1}$ via Gaussian elimination, making use of maximal pivoting.

The original matrix $\underset{\sim}{A}$ is destroyed in the process, and $\underset{\sim}{A}^{-1}$ is stored in the place of $\underset{\sim}{A}$.

MATINV also computes the determinant of $\underset{\sim}{A}$ which can be outputed for diagnostic purposes.

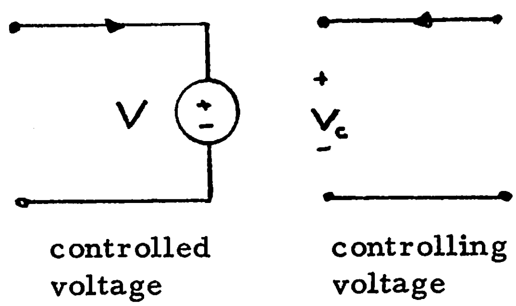
APPENDIX A

Dependent Source Modeling:

Each dependent source requires two branches for its complete specification, both of which must be a part of the network description, and hence must be read in as data.

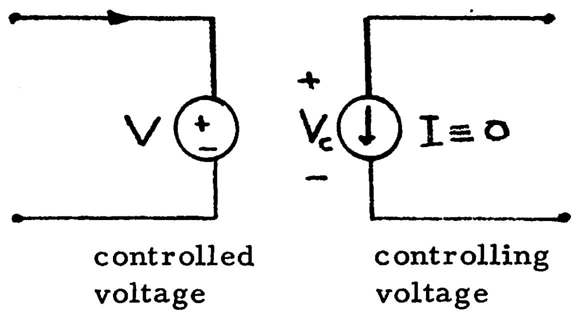
Voltage Controlled Voltage Source:

Ideal model:



$V \equiv kV_c$

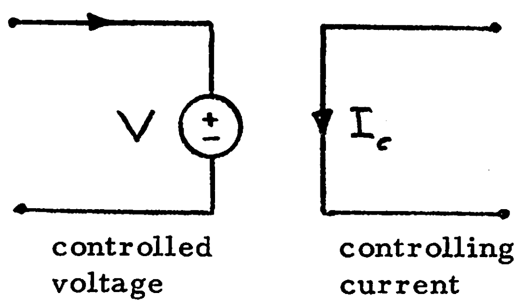
CANINE model:



$V \equiv kV_c$

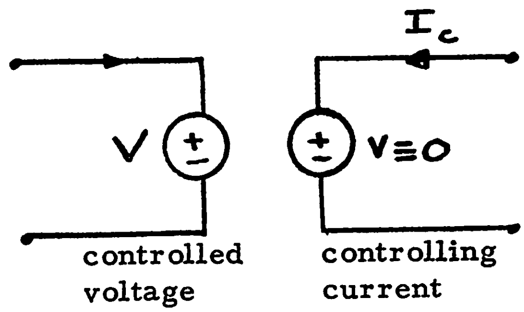
Current Controlled Voltage Source:

Ideal model:



$V \equiv kI_c$

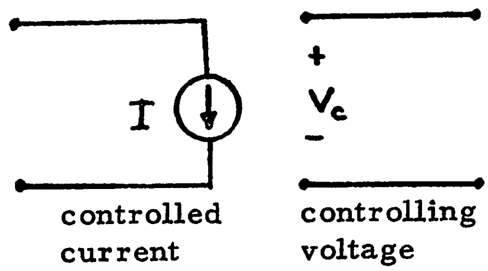
CANINE model:



$V \equiv kI_c$

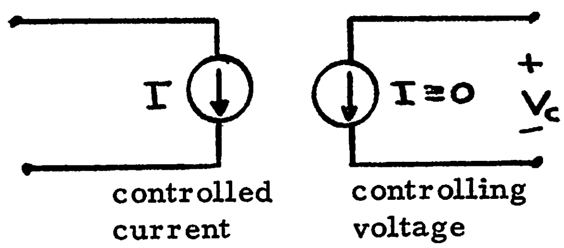
<u>Voltage Controlled Current Source:</u>

Ideal model:



$I \equiv kV_c$

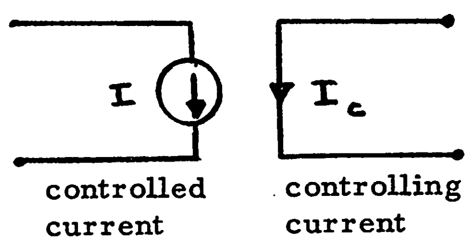
CANINE model:



$I \equiv kV_c$

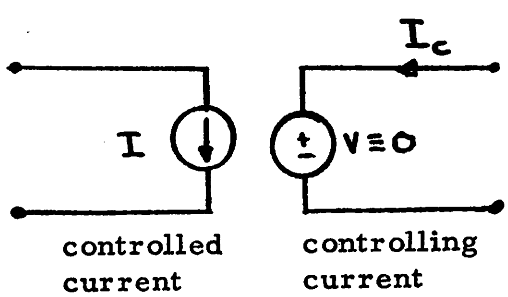
<u>Current Controlled Current Source:</u>

Ideal model:



$I \equiv kI_c$

CANINE model:



$I \equiv kI_c$

Thus we see that in all cases the controlling current is taken to be the current through a zero-valued voltage source, and the controlling voltage is taken to be the voltage across a zero-valued current source.

The coupling constants, denoted by k, are those that should be read in as corresponding elements of the VALUE array.

APPENDIX B

Data Cards for CANINE:

Notation:

I $\Rightarrow$ Integer format

E $\Rightarrow$ Exponential or floating point format

A $\Rightarrow$ Alphanumeric format

col $\Rightarrow$ Column on data card

Card # 1:

Variables read in, in sequential order

NN, number of nodes (I)

NB, number of branches (I)

TSTART, starting time (E)

TEND, end time (E)

H, time increment (E)

NSTEP, number of time iterations per output (I)

NCONT, tree and $\underset{\sim}{F}$ output control (I)

NCONT = 1 $\Rightarrow$ outputs desired

NCONT = 0 $\Rightarrow$ outputs not desired

col 1 - 5, - NN	DATA CARD # 1
col 6 - 10, - NB	
col 11 - 25, - TSTART	
col 26 - 40, - TEND	
col 41 - 55, - H	
col 56 - 60, - NSTEP	
col 80, - NCONT	

Card # 2:

Variables read in, in sequential order

NGRAPH, number of graphical outputs (I)

NALLOUT, control variable for use of ALLOUT subroutine (I)

NALLOUT = 1 $\Rightarrow$ use of ALLOUT is desired

NALLOUT = 0 $\Rightarrow$ use of ALLOUT is not desired

JOUT, number of outputs desired (I). To be used in conjunction with subroutine READOUT.

SCALE, scale factor (E)

eg. scale factor of 10^3 sets current unit to milliamps.

NSCALE, control variable for use of internal, automatic current scaling.

NSCALE = 1, $\Rightarrow$ use scale factor read in as SCALE

NSCALE $\neq$ 1, $\Rightarrow$ use internal, automatic current scaling algorithm

col 1 - 5, - NGRAPH		
col 6 - 10, - NALLOUT		
col 11 - 15, - JOUT	}	DATA CARD # 2
col 16 - 30, - SCALE		
col 40, - NSCALE		

Network Description Data Cards:

For each network branch, the following card (or set of cards) is needed. The order in which network branches are read in is arbitrary.

Variables read in, in sequential order

TYPE, Branch type (A)

E - independent voltage source

V - controlled voltage source

C - capacitance

L - inductance

R - resistance

I - controlled current source

J - independent current source

IBRAN, Branch number (I)

SORTYPE, Independent source type (A)

C - constant source

E - exponential source

P - periodic source

S - sinusoidal source

T - time-varying

CONTYPE, dependent source controlling type (A)

V - voltage controlled

I - current controlled

KONBRAN, dependent source controlling branch (I)

LEAV, node which branch leaves (I)

LENT, node which branch enters (I)

NCARDS, a flag signaling that more data pertaining to this branch needs to be read in (independent sources only). (I)

NCARDS = 0 $\Rightarrow$ no more data needed

NCARDS $\neq$ 0 $\Rightarrow$ more data needed

VALUE, value of branch element (E)

— Resistances in ohms

— Inductances in henrys

— Capacitances in farads

— Dependent source coupling constants in ohms, mhos, or unitless (see Appendix A)

COND, for inductances, the initial current

for capacitances, the initial voltage

for constant independent sources, the source value (in volts or amperes). (E)

col 1 - TYPE col 2 - 4, - IBRAN col 5 - SORTYPE col 6 - CONTYPE col 7 - 9, - KONBRAN col 11 - 12, - LEAV col 14 - 15, - LENT col 17 - 18, - NCARDS col 21 - 35, - VALUE col 36 - 50, - COND	DATA CARD FOLLOWED BY OTHER RELATED CARDS IF NCARDS $\neq$ 0

If NCARDS $\neq$ 0, we need the following card (s), to describe the corresponding nonconstant independent source.

If SORTYPE = S, ie a sinusoidal source, the signal is assumed to be of the form

$$A \cdot \sin (w * t - \phi)$$

one card is required to describe the above

col 1 - 10, signal amplitude A, in volts or amperes (E)

col 11 - 20, signal frequency w, in radians/sec (E)

col 21 - 30, signal phase ϕ, at initial time (TSTART), (E)

If SORTYPE = E, ie an exponential source, the singal is assumed to be of the form

$$A\ell^{\gamma(t - \phi)}$$

one card is required to describe the above

col 1 - 10, signal amplitude, in volts or amperes (E)

col 11 - 20, time constant γ (sec^{-1}), (E)

col 21 - 30, signal phase ϕ, at t = TSTART. (E)

If SORTYPE = T or P, iℓ a time varying or periodic source, we need the following set of cards.

a) card # 1, col 1-5, number of time points (I)

b) as many cards as needed to specify the source values at the time points, allowing for 8 source values/data card, each being alotted 10 columns of space (E).

c) as many cards as needed to specify the time points, allowing for 8 time points /data card, each being alotted 10 columns of space (E).

<u>OUTPUT specifications:</u>

Only one type of output is allowed for any one network analysis, ie, only one of NGRAPH, NALLOUT and JOUT can be nonzero.

The following data cards follow immediately after the network description data cards.

IF NGRAPH $\neq 0$, we need NGRAPH (≤ 5) data cards with the following information

col 1 - 5, branch number (I)

col 10, output type desired (I)

1 $\Rightarrow$ current desired

0 $\Rightarrow$ voltage desired

If JOUT $\neq 0$, we need JOUT (≤ 200) data cards with the following information

col 1 - 5, branch number (I)

col 10, output type desired (I)

1 $\Rightarrow$ current desired

0 $\Rightarrow$ voltage desired

If NALLOUT $\neq 0$, no other data cards are required.

Note that when one or more networks are analyzed in one batch, the last data card of the batch should be a blank card.

APPENDIX C

Sample Problems:

Notation:

— Circled numbers indicate node numbers

— Uncircled numbers indicate branch numbers

— R's indicate resistances, in ohms

— L's indicate inductances, in henrys

— C's indicate capacitances, in farads

— NN is the number of nodes

— NB is the number of branches

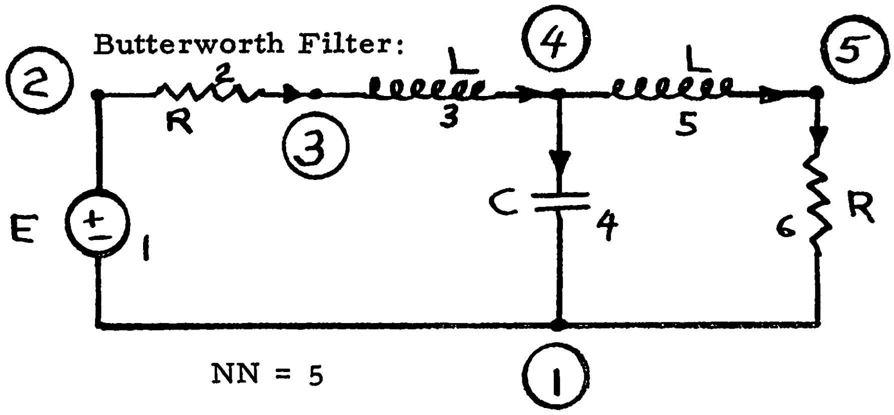
— H is the integration time step, in seconds

— TSTART is the starting time, in seconds

— TEND is the final time, in seconds

Sample Problem # 1:

Butterworth Filter:



$R = 10^3$

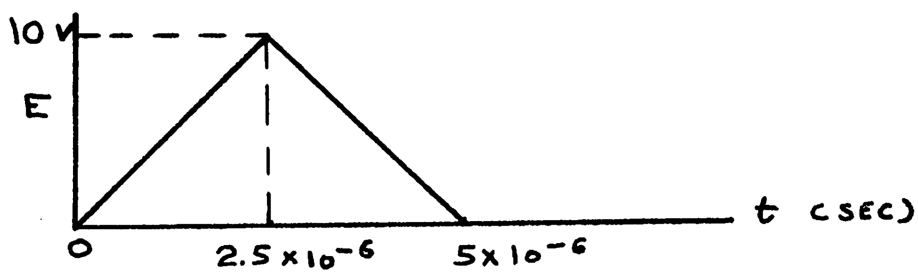
$L = 0.01$

$C = 2 \times 10^{-8}$

NN = 5

NB = 6

The input E is specified to be



Specifications:

— Initial time = 0 (TSTART = 0.0)

— Final time = 10^{-4} (TEND = 0.0001)

— Time increment = 2×10^{-6} (H = 0.000002)

— Desire results, at each time point, to be outputed (NSTEP = 1)

— Desire tree and $\underset{\sim}{F}$ to be outputed (NCONT = 1)

— Desire current to be in millamps (SCALE = 10^3 and NSCALE = 1)

— Desire the following graphical outputs

Branch 1 voltage Branch 2 current Branch 6 voltage	(NGRAPH = 3)

Specify zero intial conditions

The required data cards, for this problem, may be seen on the following page.

The central processor time, for this problem, was 2.86 seconds.

Comments on Sample Problem # 1:

The proper tree selected at the initial time point coincides with the optimal tree picked at the second time point. Thus $\underset{\sim}{F}$ is computed only once and hence, is outputed only once.

DATA CARDS : PROBLEM #1

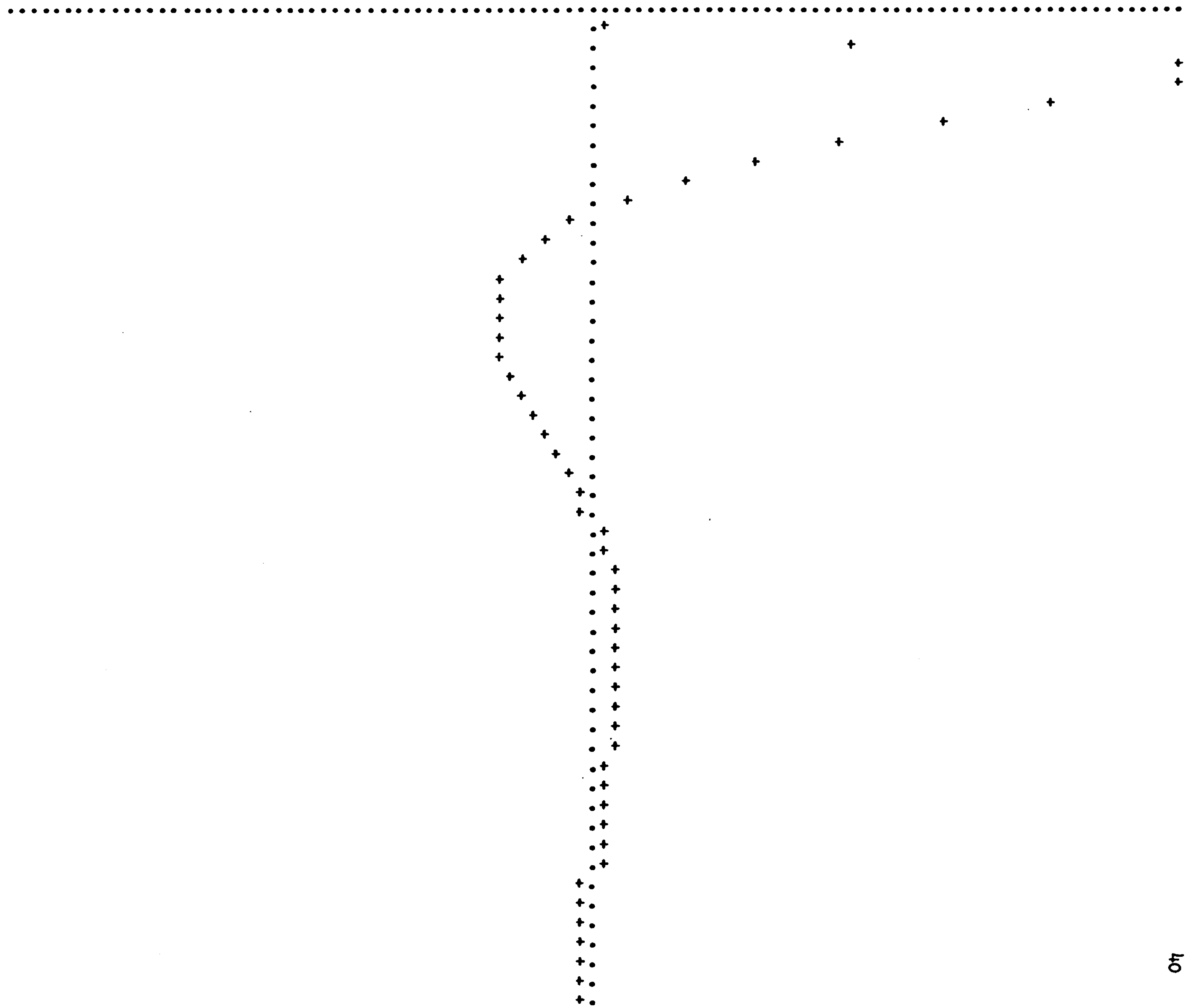
```
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
    5    6            0.0         0.0001        0.2E-05    1                   1
    3                    1000.0         1
E  1T      2  1  3
    3
       0.0      10.0       0.0
       0.0  0.25E-05   0.5E-05
R  2       2  3                 1000.0
L  3       3  4                   0.01
C  4       4  1                0.2E-07
L  5       4  5                   0.01
R  6       5  1                 1000.0
    1    0
    2    1
    6    0
```

BRANCH 1 VOLTAGE

TIME	VALUE
0.	0.
.2000E-05	.8000E+01
.4000E-05	.4000E+01
.6000E-05	0.
.8000E-05	0.
.1000E-04	0.
.1200E-04	0.
.1400E-04	0.
.1600E-04	0.
.1800E-04	0.
.2000E-04	0.
.2200E-04	0.
.2400E-04	0.
.2600E-04	0.
.2800E-04	0.
.3000E-04	0.
.3200E-04	0.
.3400E-04	0.
.3600E-04	0.
.3800E-04	0.
.4000E-04	0.
.4200E-04	0.
.4400E-04	0.
.4600E-04	0.
.4800E-04	0.
.5000E-04	0.
.5200E-04	0.
.5400E-04	0.
.5600E-04	0.
.5800E-04	0.
.6000E-04	0.
.6200E-04	0.
.6400E-04	0.
.6600E-04	0.
.6800E-04	0.
.7000E-04	0.
.7200E-04	0.
.7400E-04	0.
.7600E-04	0.
.7800E-04	0.
.8000E-04	0.
.8200E-04	0.
.8400E-04	0.
.8600E-04	0.
.8800E-04	0.
.9000E-04	0.
.9200E-04	0.
.9400E-04	0.
.9600E-04	0.
.9800E-04	0.
.1000E-03	0.

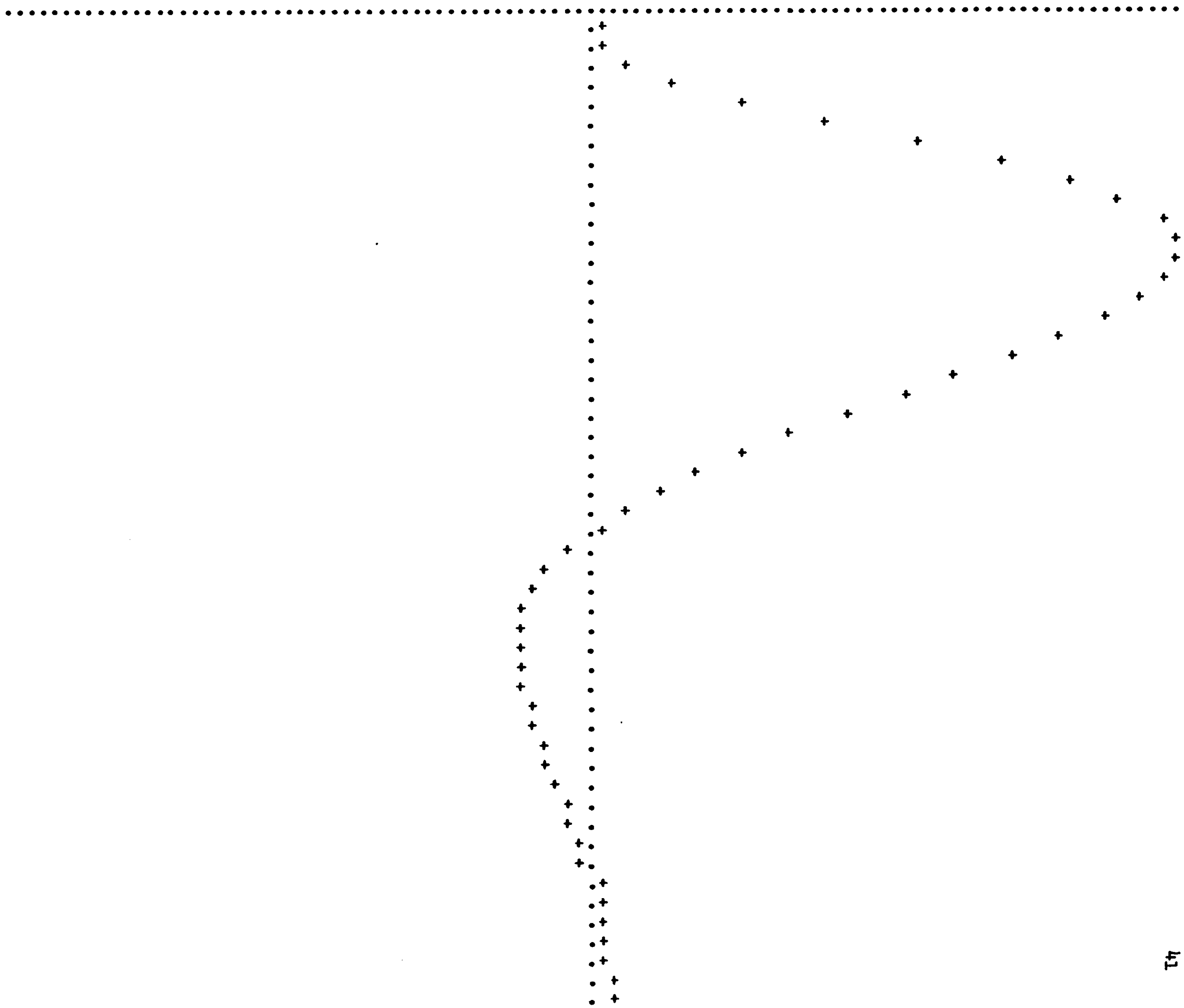
BRANCH 2 CURRENT

TIME	VALUE
0.	0.
.2000E-05	.7240E-03
.4000E-05	.1666E-02
.6000E-05	.1684E-02
.8000E-05	.1307E-02
.1000E-04	.9771E-03
.1200E-04	.6925E-03
.1400E-04	.4517E-03
.1600E-04	.2522E-03
.1800E-04	.9122E-04
.2000E-04	-.3454E-04
.2200E-04	-.1286E-03
.2400E-04	-.1947E-03
.2600E-04	-.2357E-03
.2800E-04	-.2583E-03
.3000E-04	-.2633E-03
.3200E-04	-.2550E-03
.3400E-04	-.2366E-03
.3600E-04	-.2112E-03
.3800E-04	-.1811E-03
.4000E-04	-.1487E-03
.4200E-04	-.1157E-03
.4400E-04	-.8358E-04
.4600E-04	-.5357E-04
.4800E-04	-.2646E-04
.5000E-04	-.2818E-05
.5200E-04	.1705E-04
.5400E-04	.3304E-04
.5600E-04	.4522E-04
.5800E-04	.5378E-04
.6000E-04	.5903E-04
.6200E-04	.6135E-04
.6400E-04	.6116E-04
.6600E-04	.5889E-04
.6800E-04	.5499E-04
.7000E-04	.4987E-04
.7200E-04	.4392E-04
.7400E-04	.3751E-04
.7600E-04	.3094E-04
.7800E-04	.2446E-04
.8000E-04	.1829E-04
.8200E-04	.1250E-04
.8400E-04	.7487E-05
.8600E-04	.3042E-05
.8800E-04	-.7020E-06
.9000E-04	-.3737E-05
.9200E-04	-.6084E-05
.9400E-04	-.7783E-05
.9600E-04	-.8891E-05
.9800E-04	-.9475E-05
.1000E-03	-.9509E-05



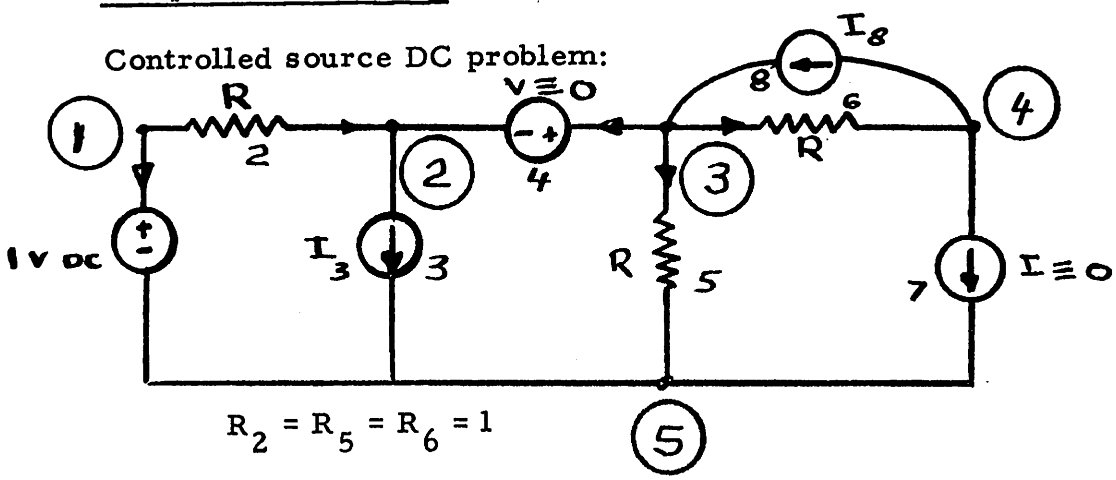
BRANCH 6 VOLTAGE

TIME	VALUE
0.	0.
.2000E-05	.3276E-02
.4000E-05	.1999E-01
.6000E-05	.5929E-01
.8000E-05	.1191E+00
.1000E-04	.1898E+00
.1200E-04	.2621E+00
.1400E-04	.3294E+00
.1600E-04	.3869E+00
.1800E-04	.4317E+00
.2000E-04	.4624E+00
.2200E-04	.4787E+00
.2400E-04	.4811E+00
.2600E-04	.4710E+00
.2800E-04	.4500E+00
.3000E-04	.4201E+00
.3200E-04	.3833E+00
.3400E-04	.3416E+00
.3600E-04	.2971E+00
.3800E-04	.2514E+00
.4000E-04	.2062E+00
.4200E-04	.1627E+00
.4400E-04	.1221E+00
.4600E-04	.8507E-01
.4800E-04	.5224E-01
.5000E-04	.2390E-01
.5200E-04	.1997E-03
.5400E-04	-.1893E-01
.5600E-04	-.3367E-01
.5800E-04	-.4433E-01
.6000E-04	-.5130E-01
.6200E-04	-.5503E-01
.6400E-04	-.5598E-01
.6600E-04	-.5466E-01
.6800E-04	-.5152E-01
.7000E-04	-.4703E-01
.7200E-04	-.4160E-01
.7400E-04	-.3561E-01
.7600E-04	-.2938E-01
.7800E-04	-.2319E-01
.8000E-04	-.1725E-01
.8200E-04	-.1175E-01
.8400E-04	-.6791E-02
.8600E-04	-.2473E-02
.8800E-04	.1168E-02
.9000E-04	.4118E-02
.9200E-04	.6396E-02
.9400E-04	.8038E-02
.9600E-04	.9100E-02
.9800E-04	.9646E-02
.1000E-03	.9749E-02



Sample Problem # 2:

Controlled source DC problem:



$R_2 = R_5 = R_6 = 1$

NN = 5

NB = 8

Branch 3 is a current controlled current source, controlled by branch 4

$$I_3 = 5 * I_4$$

Branch 8 is a voltage controlled current source, controlled by branch 7

$$I_8 = 3 * V_7$$

Specifications:

— Since our problem is a resistive DC network, we need the solution at only one time point. Set TSTART = 0.0 = TEND

— The time increments is arbitrary (set H = 0.0)

— Desire results at each time point to be outputed (NSTEP = 1)

— Do not desire tree and $\underset{\sim}{F}$ to be outputed (NCONT = 0)

— Desire current to be in amperes. (SCALE = 1.0 and NSCALE = 1)

— Desire all voltages and currents outputed (JOUT = 16)

The required data cards, for this problem, may be seen on the following page. The central processor time, for this problem, was 0.234 seconds.

DATA CARDS: PROBLEM #2

Cols 1–5	6	7–10	11–15	21–25	26–30	31–35	36–40	46–50	51–55	56–60
5		8		0.0			0.0		0.0	1
			16		1.0		1			
E 1C			1 5					1.0		
A 2			1 2			1.0				
I 3	I	4	2 5			5.0				
E 4C			3 2							
R 5			3 5			1.0				
R 6			3 4			1.0				
J 7C			4 5							
I 8	V	7	4 3			3.0				
1		1								
1		0								
2		1								
2		0								
3		1								
3		0								
4		1								
4		0								
5		1								
5		0								
6		1								
6		0								
7		1								

PROBLEM #2 CONT'D

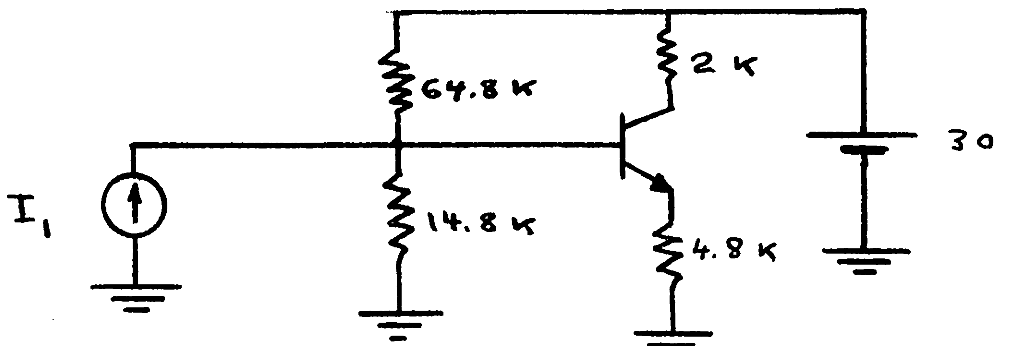
7	0
8	1
8	0

TIME = V.

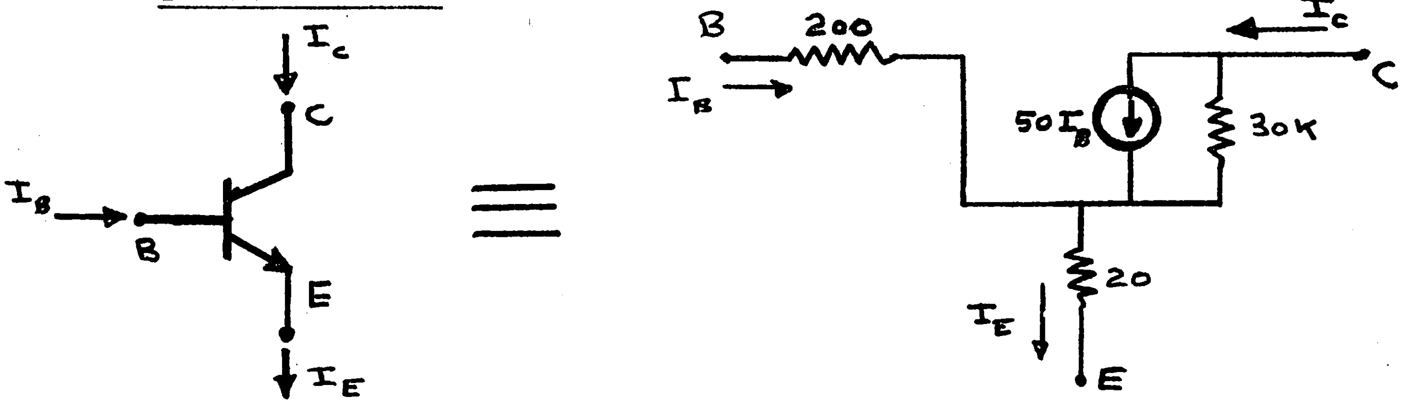
BRANCH	1	CURRENT	-.13333E+01
BRANCH	1	VOLTAGE	.10000E+01
BRANCH	2	CURRENT	.13333E+01
BRANCH	2	VOLTAGE	.13333E+01
BRANCH	3	CURRENT	.16667E+01
BRANCH	3	VOLTAGE	-.33333E+00
BRANCH	4	CURRENT	.33333E+00
BRANCH	4	VOLTAGE	0.
BRANCH	5	CURRENT	-.33333E+00
BRANCH	5	VOLTAGE	-.33333E+00
BRANCH	6	CURRENT	-.25000E+00
BRANCH	6	VOLTAGE	-.25000E+00
BRANCH	7	CURRENT	0.
BRANCH	7	VOLTAGE	-.83333E-01
BRANCH	8	CURRENT	-.25000E+00
BRANCH	8	VOLTAGE	.25000E+00

Sample Problem # 3:

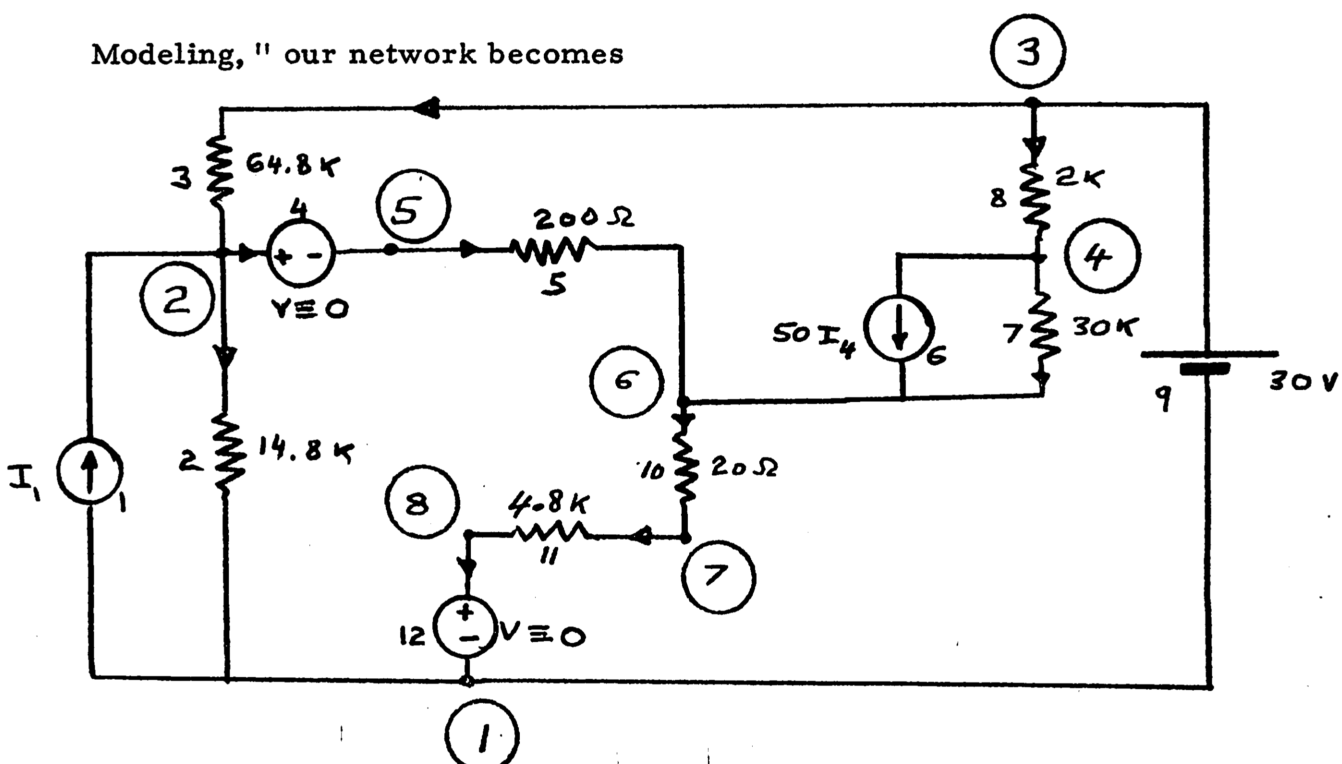
D. C. Amplifier



Transistor Model



The output will be the emitter current, and will be taken to be a current through a zero valved voltage source. Modeling the controlled source in the way described in the section on "Dependent Source Modeling," our network becomes



The approximate theoretical gain (neglecting the 30 K across the controlled current source) can be shown to be:

$$\frac{\Delta I_{12}}{\Delta I_1} \triangleq 2.4$$

about a quiescent value of I_{12} of 1.1 ma.

Specifications:

$I_1 = A \sin(1.8\,t)$

A = 0.1 milliamps

t is in seconds

— initial time = 0 (TSTART = 0.0)

— final time = 5 (TEND = 5.0)

— time increment = 0.1 (H = 0.1)

— desire results, at each time point, to be outputed (NSTEP = 1)

— desire tree and F to be outputed (NCONT = 1)

— desire internal, automatic current scaling (NSCALE ≠ 1)

— desire the following graphical outputs

Branch 1 current
Branch 4 current
Branch 12 current
Branch 6 current
} (NGRAPH = 4)

The required data cards, for this problem, may be seen on the following page.

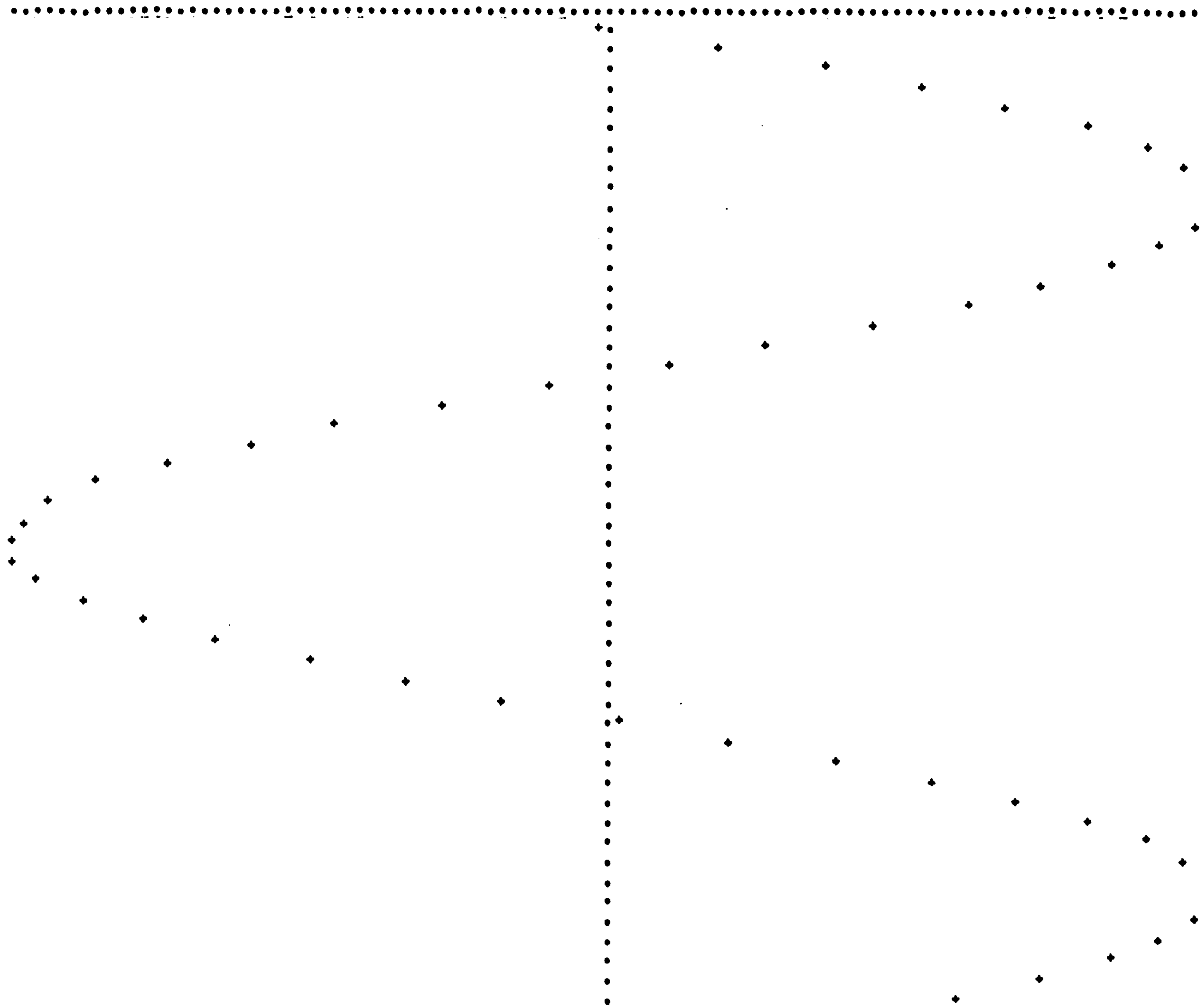
The central processor time, for this problem, was 8.48 seconds.

DATA CARDS: PROBLEM # 3

1–5	6	7–10	11–15	16–20	21–25	26–30	31–35	36–40	41–50	51–55	56–60	61–80
8		12			0.0			5.0		0.1	1	1
4								0				
J 1S			1 2	1								
0	.	0001		1.8		0.0						
E 4C			2 5									
R 5			5 6				2000.0					
R 10			6 7				20.0					
R 11			7 8				4800.0					
R 2			2 1				14800.0					
R 3			3 2				64800.0					
L 6	I	4	4 6				50.0					
R 7			4 6				30000.0					
E 9C			3 1						30.0			
R 8			3 4				2000.0					
E 12C			8 1									
1		1										
4		1										
6		1										
12		1										

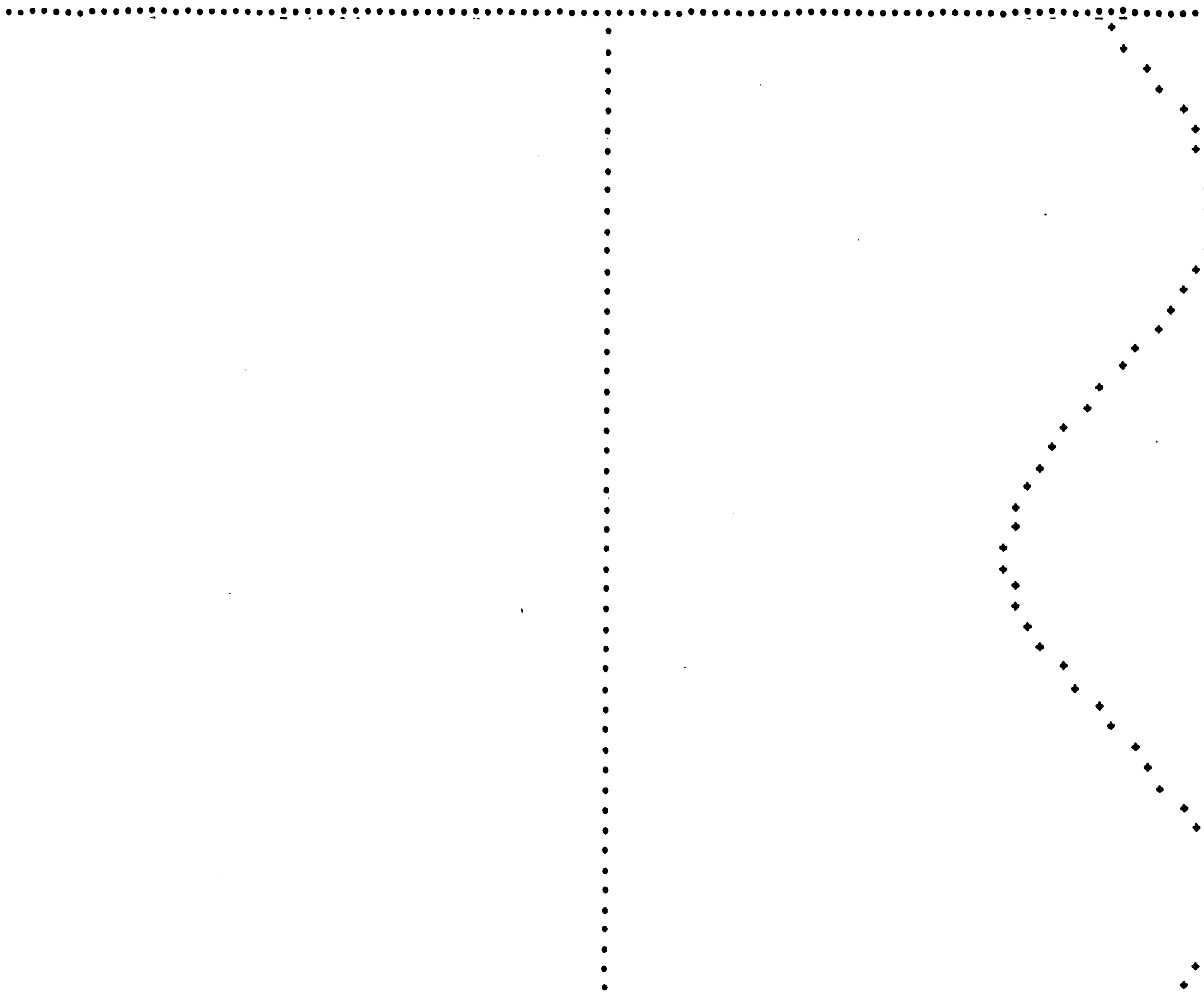
BRANCH 1 CURRENT

TIME	VALUE
0.	-.9002E-18
.1000E+00	.1790E-04
.2000E+00	.3523E-04
.3000E+00	.5141E-04
.4000E+00	.6594E-04
.5000E+00	.7833E-04
.6000E+00	.8820E-04
.7000E+00	.9521E-04
.8000E+00	.9915E-04
.9000E+00	.9988E-04
.1000E+01	.9738E-04
.1100E+01	.9174E-04
.1200E+01	.8314E-04
.1300E+01	.7185E-04
.1400E+01	.5823E-04
.1500E+01	.4274E-04
.1600E+01	.2586E-04
.1700E+01	.8150E-05
.1800E+01	-.9825E-05
.1900E+01	-.2748E-04
.2000E+01	-.4425E-04
.2100E+01	-.5959E-04
.2200E+01	-.7301E-04
.2300E+01	-.8406E-04
.2400E+01	-.9240E-04
.2500E+01	-.9775E-04
.2600E+01	-.9995E-04
.2700E+01	-.9891E-04
.2800E+01	-.9468E-04
.2900E+01	-.8739E-04
.3000E+01	-.7728E-04
.3100E+01	-.6467E-04
.3200E+01	-.4996E-04
.3300E+01	-.3365E-04
.3400E+01	-.1625E-04
.3500E+01	.1681E-05
.3600E+01	.1955E-04
.3700E+01	.3680E-04
.3800E+01	.5285E-04
.3900E+01	.6719E-04
.4000E+01	.7937E-04
.4100E+01	.8898E-04
.4200E+01	.9571E-04
.4300E+01	.9935E-04
.4400E+01	.9978E-04
.4500E+01	.9699E-04
.4600E+01	.9106E-04
.4700E+01	.8219E-04
.4800E+01	.7067E-04
.4900E+01	.5686E-04



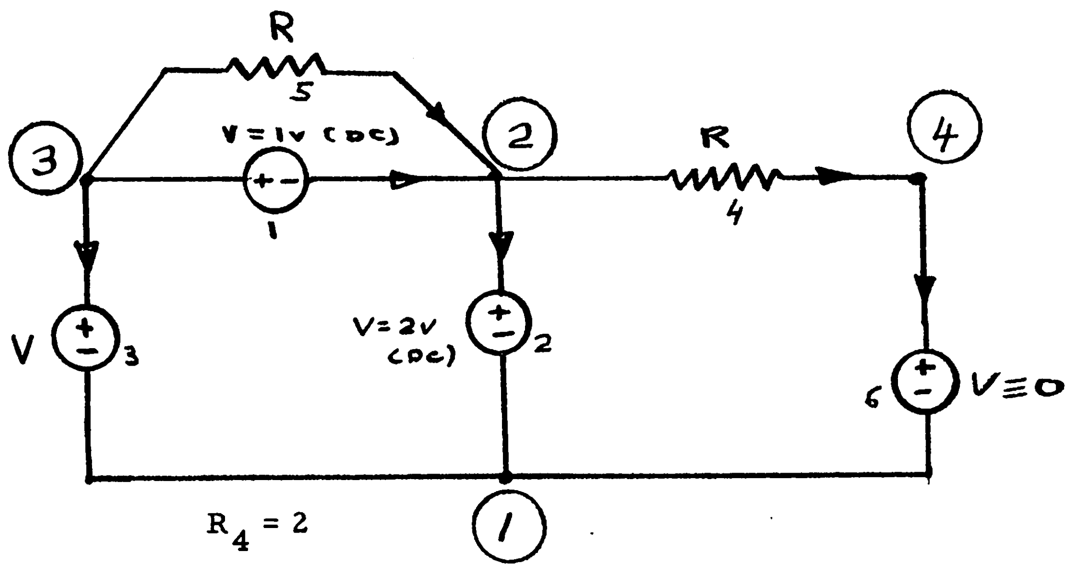
BRANCH 12 CURRENT

TIME	VALUE
0.	.1138E-02
.1000E+00	.1180E-02
.2000E+00	.1221E-02
.3000E+00	.1259E-02
.4000E+00	.1293E-02
.5000E+00	.1322E-02
.6000E+00	.1345E-02
.7000E+00	.1362E-02
.8000E+00	.1371E-02
.9000E+00	.1373E-02
.1000E+01	.1367E-02
.1100E+01	.1354E-02
.1200E+01	.1333E-02
.1300E+01	.1307E-02
.1400E+01	.1275E-02
.1500E+01	.1238E-02
.1600E+01	.1198E-02
.1700E+01	.1157E-02
.1800E+01	.1114E-02
.1900E+01	.1073E-02
.2000E+01	.1033E-02
.2100E+01	.9971E-03
.2200E+01	.9655E-03
.2300E+01	.9395E-03
.2400E+01	.9199E-03
.2500E+01	.9072E-03
.2600E+01	.9021E-03
.2700E+01	.9045E-03
.2800E+01	.9145E-03
.2900E+01	.9317E-03
.3000E+01	.9555E-03
.3100E+01	.9852E-03
.3200E+01	.1020E-02
.3300E+01	.1058E-02
.3400E+01	.1099E-02
.3500E+01	.1141E-02
.3600E+01	.1184E-02
.3700E+01	.1224E-02
.3800E+01	.1262E-02
.3900E+01	.1296E-02
.4000E+01	.1324E-02
.4100E+01	.1347E-02
.4200E+01	.1363E-02
.4300E+01	.1372E-02
.4400E+01	.1373E-02
.4500E+01	.1366E-02
.4600E+01	.1352E-02
.4700E+01	.1331E-02
.4800E+01	.1304E-02
.4900E+01	.1271E-02



Sample Problem # 4:

Compatible voltage source loop



$R_4 = 2$

$R_5 = 1$

NN = 4

NB = 6

Branch 3 is a current controlled voltage source, controlled by branch 6

$$V_3 = 3 * I_6$$

Specifications:

— since we have a resistive DC problem, we need solution at only one time point.

— the time increment is arbitrary, (H = 0.0)

— desire results, at each time point, to be outputed. (NSTEP = 1)

— do not desire tree and $\underset{\sim}{F}$, to be outputed (NCONT = 0)

— desire automatic internal scaling (NSCALE $\neq$ 1)

— desire all branch currents and voltages to be outputed.

The required data cards, for this problem, may be seen on the following page.

The central processor time, for this problem, was 0.23 seconds.

Comments on Sample Problem # 4:

Note that when dependent and indpendent voltage source loops exist in a network, it is of paramount importance for them to be "compatible," that Kirchhoff's voltage law be satisfied around that loop.

Incompatible loops, of the above type, will yield incorrect executions of CANINE. The same holds for cutsets of independent and dependent current sources.

DATA CARDS : PROBLEM # 4

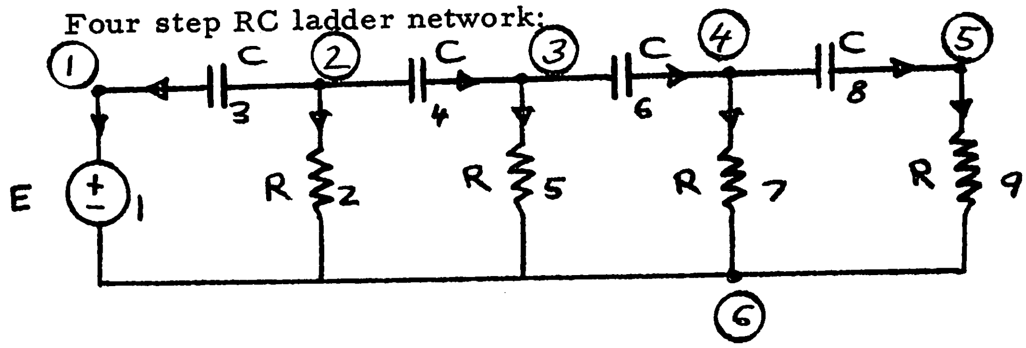
```
    4    6          0.0            0.0           0.0    1
             12
E  1C      3  2                                 1.0
E  2C      2  1                                 2.0
V  3 I  6  3  1                 3.0
R  4       2  4                 2.0
R  5       3  2                 1.0
E  6C      4  1
    1    1
    2    1
    3    1
    4    1
    5    1
    6    1
    1    0
    2    0
    3    0
    4    0
    5    0
    6    0
```

TIME = 0.

BRANCH	1	CURRENT	-.10000E+01
BRANCH	2	CURRENT	-.10000E+01
BRANCH	3	CURRENT	.24136-225
BRANCH	4	CURRENT	.10000E+01
BRANCH	5	CURRENT	.10000E+01
BRANCH	6	CURRENT	.10000E+01
BRANCH	1	VOLTAGE	.10000E+01
BRANCH	2	VOLTAGE	.20000E+01
BRANCH	3	VOLTAGE	.30000E+01
BRANCH	4	VOLTAGE	.20000E+01
BRANCH	5	VOLTAGE	.10000E+01
BRANCH	6	VOLTAGE	0.

Sample Problem # 5:

Four step RC ladder network:



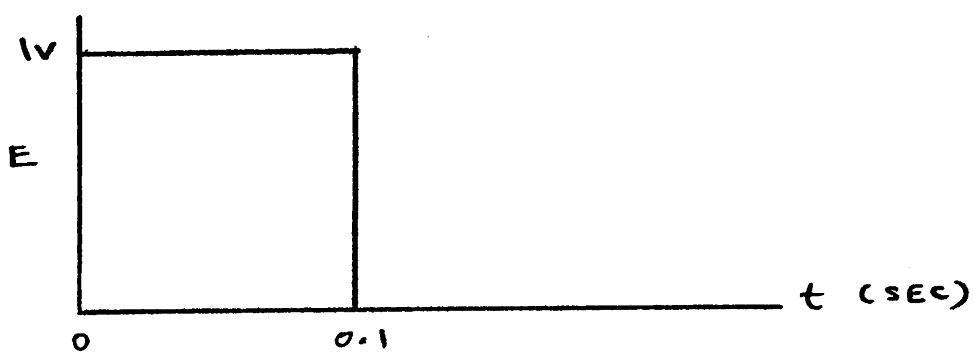
R = 10^3

C = 10^{-5}

NN = 6

NB = 9

The input E is specified as follows:



Specifications:

— initial time = 0 (TSTART = 0.0)

— final time = 50 milliseconds (TEND = 0.05)

— time increment = 1 millisecond (H = 0.001)

— desire results at each time point to be outputed (NSTEP = 1)

— do not desire tree information and F to be outputed (NCONT = 0)

— desire internal automatic current scaling (NSCALE $\neq$ 1)

— zero initial conditions

— desire the following graphical outputs

Branch 1 voltage } (NGRAPH = 2)
Branch 5 voltage }

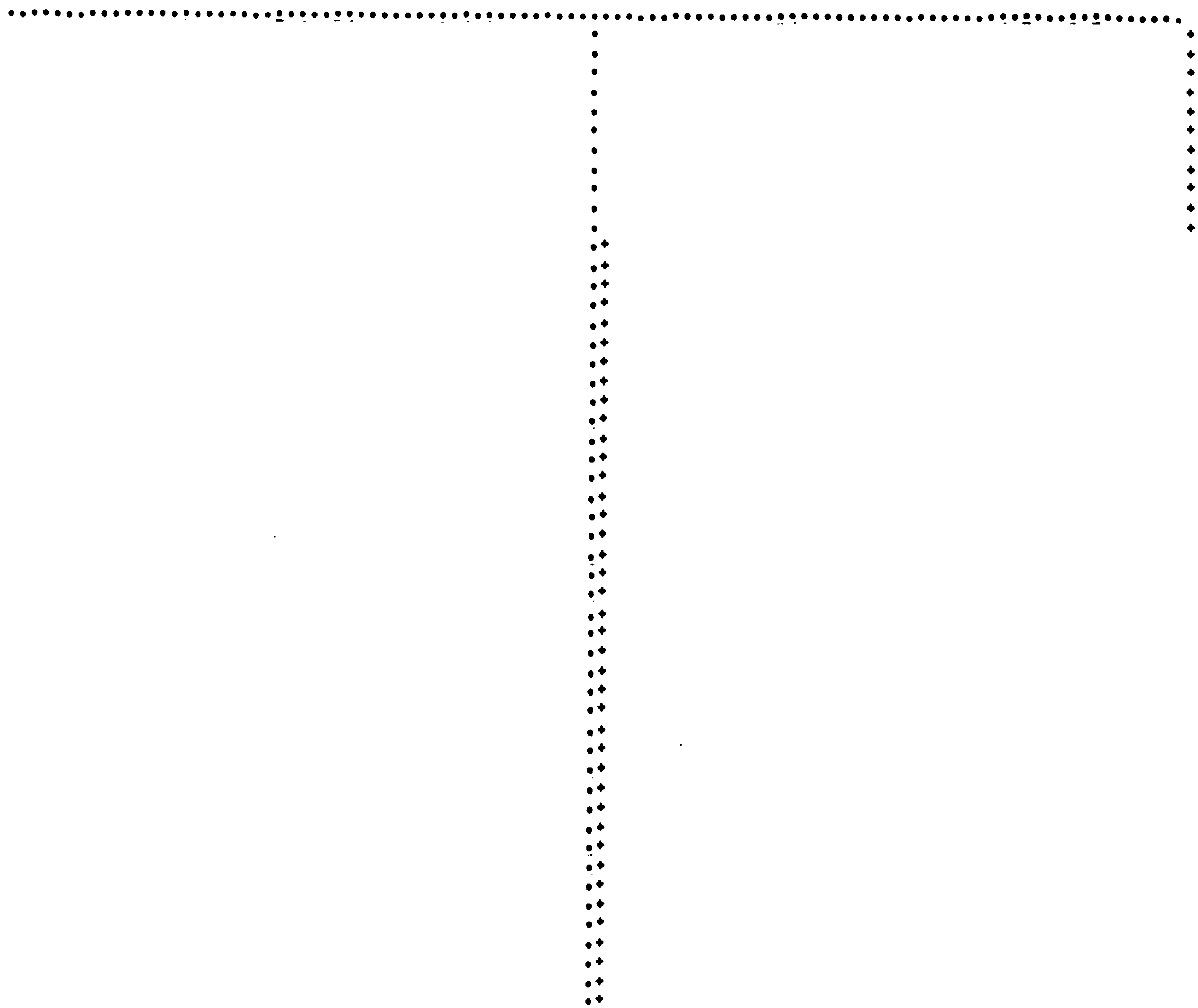
The required data cards for this problem may be seen on the following page. The central processor time, for this problem, was 4.17 seconds.

DATA CARDS: PROBLEM #5

```
         1111111111222222222233333333334444444444555555555566666666667777777777
1234567890123456789012345678901234567890123456789012345678901234567890123456789
    6    9            0.0           0.05          0.001    1
    2
E  1T      1  6  1                             1.0
    3
       1.0       1.0       0.0
       0.0      0.01      0.011
R  9       5  6              1000.0
R  7       4  6              1000.0
R  2       2  6              1000.0
R  5       3  6              1000.0
C  3       2  1             0.00001
C  4       2  3             0.00001
C  6       3  4             0.00001
C  8       4  5             0.00001
    1    0
    5    0
```

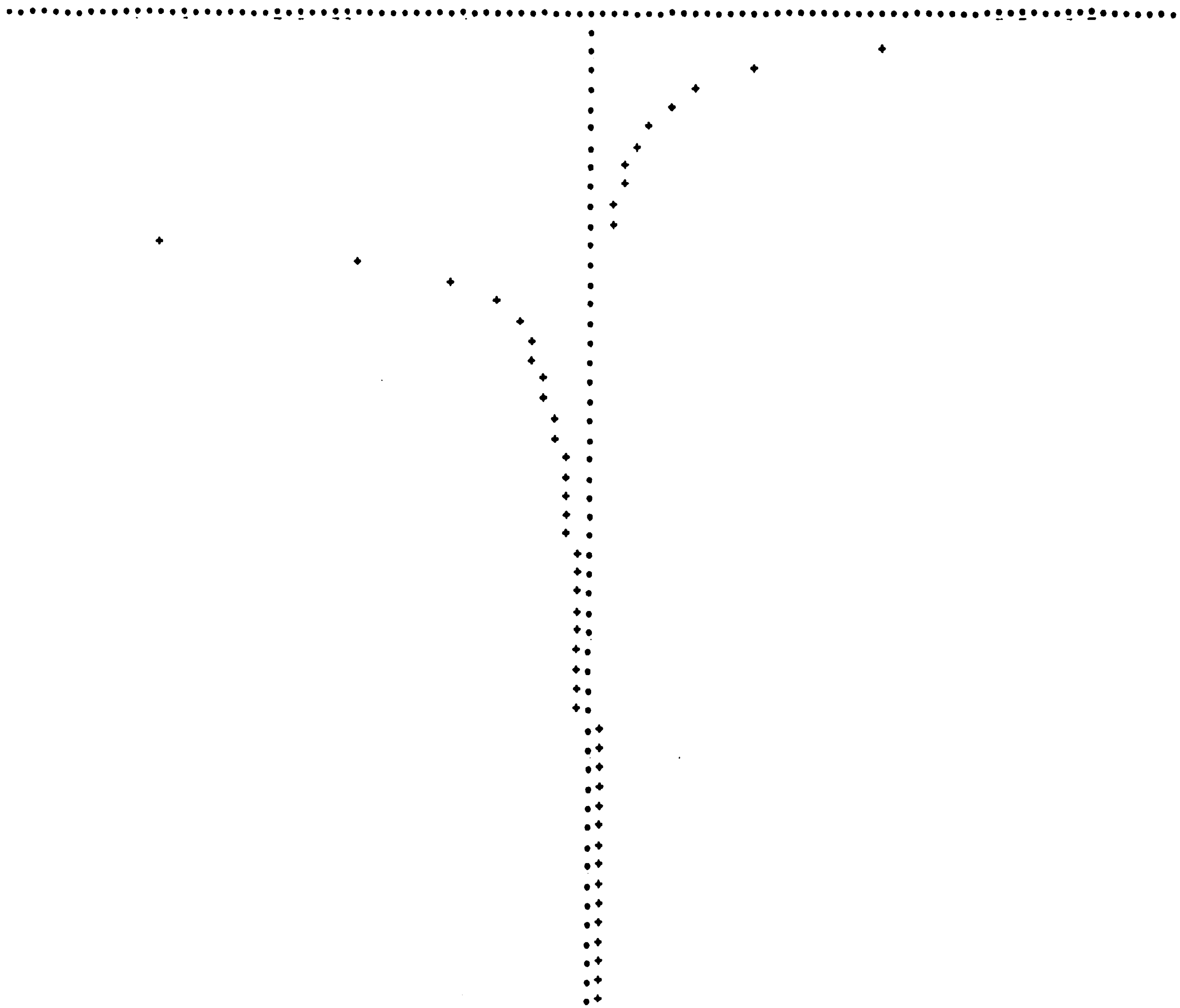
BRANCH 1 VOLTAGE

TIME	VALUE
0.	.1000E+01
.1000E-02	.1000E+01
.2000E-02	.1000E+01
.3000E-02	.1000E+01
.4000E-02	.1000E+01
.5000E-02	.1000E+01
.6000E-02	.1000E+01
.7000E-02	.1000E+01
.8000E-02	.1000E+01
.9000E-02	.1000E+01
.1000E-01	.1000E+01
.1100E-01	0.
.1200E-01	0.
.1300E-01	0.
.1400E-01	0.
.1500E-01	0.
.1600E-01	0.
.1700E-01	0.
.1800E-01	0.
.1900E-01	0.
.2000E-01	0.
.2100E-01	0.
.2200E-01	0.
.2300E-01	0.
.2400E-01	0.
.2500E-01	0.
.2600E-01	0.
.2700E-01	0.
.2800E-01	0.
.2900E-01	0.
.3000E-01	0.
.3100E-01	0.
.3200E-01	0.
.3300E-01	0.
.3400E-01	0.
.3500E-01	0.
.3600E-01	0.
.3700E-01	0.
.3800E-01	0.
.3900E-01	0.
.4000E-01	0.
.4100E-01	0.
.4200E-01	0.
.4300E-01	0.
.4400E-01	0.
.4500E-01	0.
.4600E-01	0.
.4700E-01	0.
.4800E-01	0.
.4900E-01	0.
.5000E-01	0.



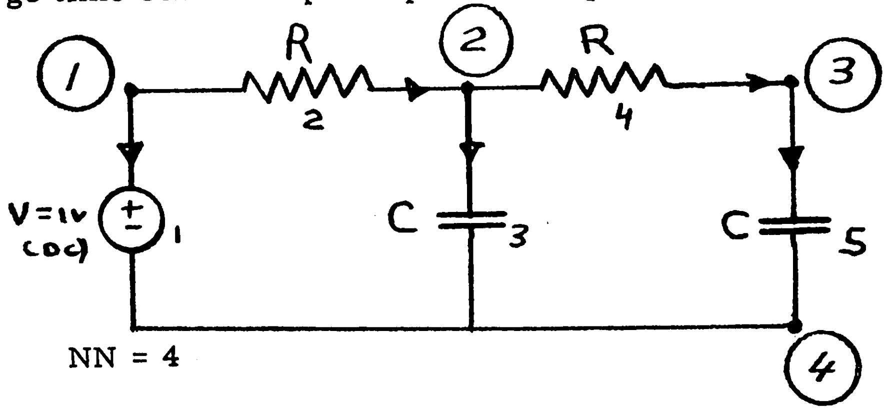
BRANCH 5 VOLTAGE

TIME	VALUE
0.	.1000E+01
.1000E-02	.4983E+00
.2000E-02	.2778E+00
.3000E-02	.1750E+00
.4000E-02	.1222E+00
.5000E-02	.9125E-01
.6000E-02	.7032E-01
.7000E-02	.5446E-01
.8000E-02	.4151E-01
.9000E-02	.3051E-01
.1000E-01	.2098E-01
.1100E-01	-.7365E+00
.1200E-01	-.3827E+00
.1300E-01	-.2274E+00
.1400E-01	-.1552E+00
.1500E-01	-.1182E+00
.1600E-01	-.9647E-01
.1700E-01	-.8171E-01
.1800E-01	-.7043E-01
.1900E-01	-.6113E-01
.2000E-01	-.5312E-01
.2100E-01	-.4609E-01
.2200E-01	-.3985E-01
.2300E-01	-.3428E-01
.2400E-01	-.2932E-01
.2500E-01	-.2489E-01
.2600E-01	-.2094E-01
.2700E-01	-.1742E-01
.2800E-01	-.1428E-01
.2900E-01	-.1150E-01
.3000E-01	-.9026E-02
.3100E-01	-.6835E-02
.3200E-01	-.4898E-02
.3300E-01	-.3188E-02
.3400E-01	-.1682E-02
.3500E-01	-.3601E-03
.3600E-01	.7974E-03
.3700E-01	.1807E-02
.3800E-01	.2685E-02
.3900E-01	.3444E-02
.4000E-01	.4098E-02
.4100E-01	.4656E-02
.4200E-01	.5131E-02
.4300E-01	.5529E-02
.4400E-01	.5861E-02
.4500E-01	.6133E-02
.4600E-01	.6351E-02
.4700E-01	.6523E-02
.4800E-01	.6653E-02
.4900E-01	.6745E-02
.5000E-01	.6806E-02



Sample Problem # 6:

Large time constant spread problem - quick response:



NN = 4

NB = 5

We desire the step response associated with the small time constant.

Specifications:

— initial time = 0 (TSTART = 0.0)

— final time = 50 microsec. (TEND = 5×10^{-5})

— time increment = 1 microsec. (H = 10^{-6})

— desire results, at each time point, to be outputed (NSTEP = 1)

— do not desire tree and F to be outputed (NCONT $\neq$ 1)

— desire internal automatic current scaling (NSCALE $\neq$ 1)

— zero initial conditions

— desire the following graphical outputs

Branch 3 voltage
Branch 4 current
Branch 2 current
Branch 5 voltage

(NGRAPH = 4)

The data cards, for this problem, may be seen on the following page.

The central processor, time for this problem, was 2.96 seconds.

Comments on Sample Problem # 6:

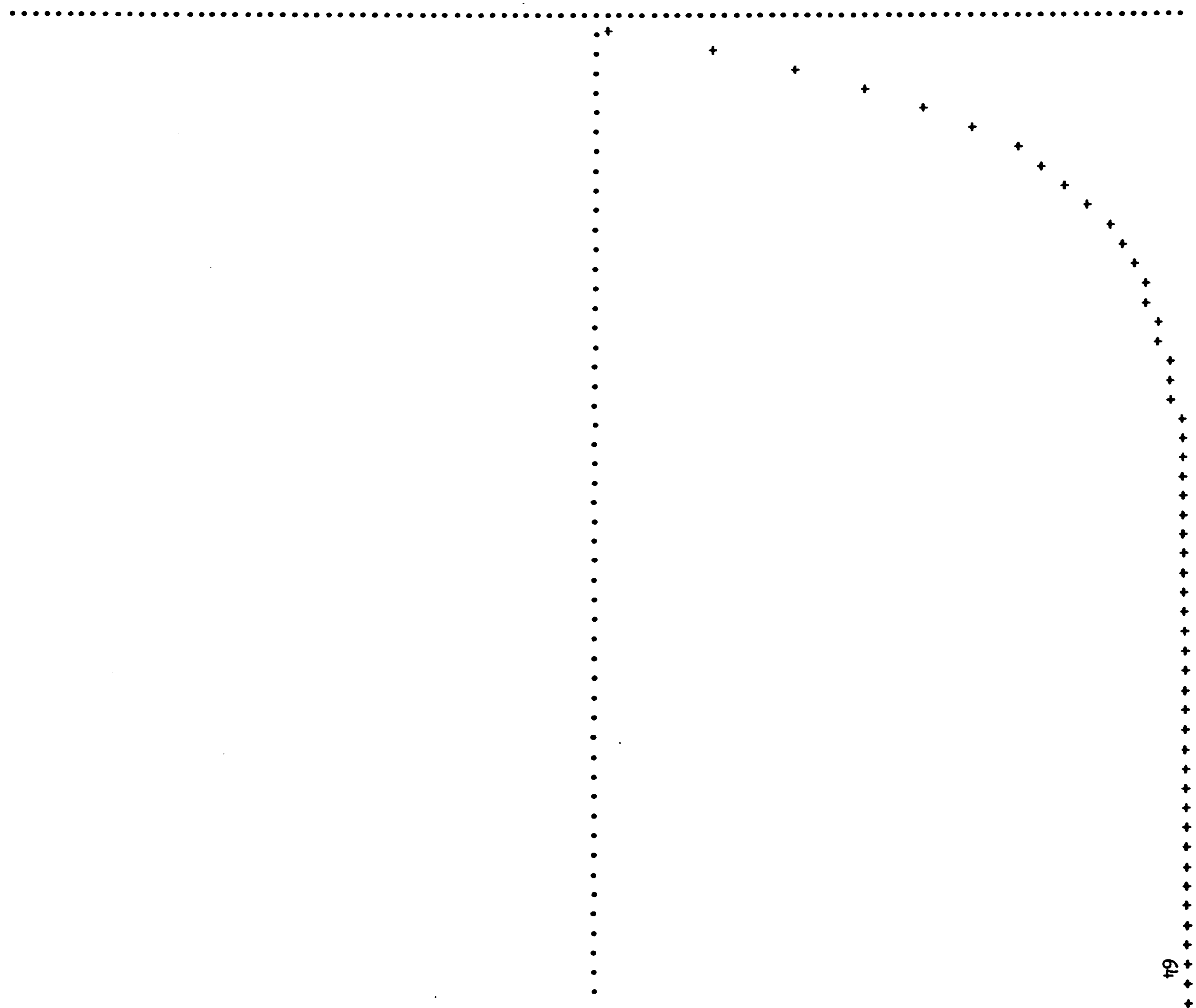
The quick response steady state voltage across (3 is 0.5 volts, and, as will be noted in problem # 7, this value must be used as the initial conditions on C_3 when the slow time response is desired.

DATA CARDS : PROBLEM #6

```
    4    5            0.0        0.5E-04        1.0E-06    1
    4
E  1C      1  4                                1.0
R   2      1  2              1000.0
C   3      2  4             1.0E-08
R   4      2  3              1000.0
C   5      3  4                 0.1
    3    0
    4    1
    2    1
    5    0
```

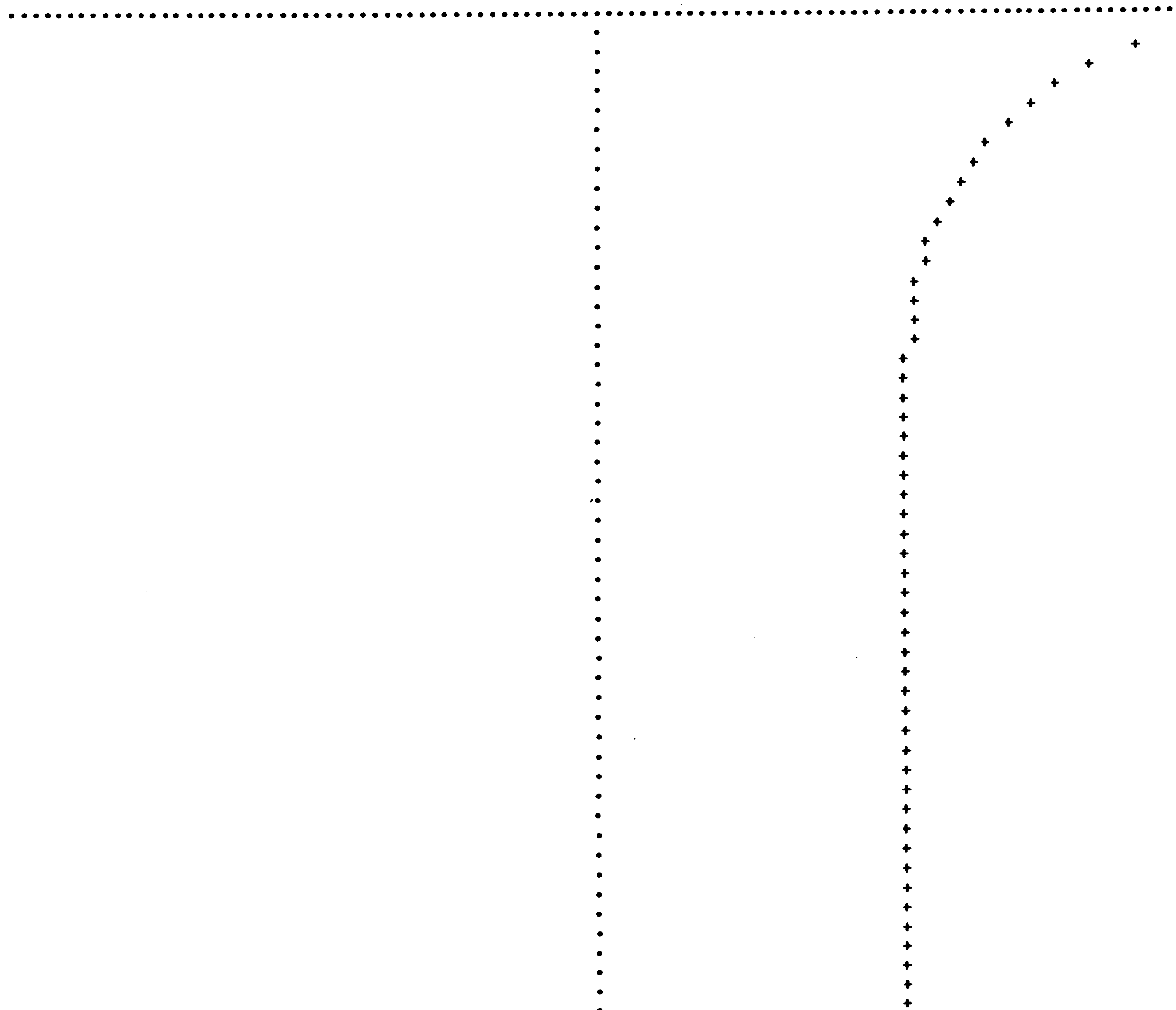
BRANCH 3 VOLTAGE

TIME	VALUE
0.	0.
.1000E-05	.9091E-01
.2000E-05	.1653E+00
.3000E-05	.2261E+00
.4000E-05	.2759E+00
.5000E-05	.3167E+00
.6000E-05	.3500E+00
.7000E-05	.3773E+00
.8000E-05	.3996E+00
.9000E-05	.4178E+00
.1000E-04	.4328E+00
.1100E-04	.4450E+00
.1200E-04	.4550E+00
.1300E-04	.4632E+00
.1400E-04	.4699E+00
.1500E-04	.4754E+00
.1600E-04	.4798E+00
.1700E-04	.4835E+00
.1800E-04	.4865E+00
.1900E-04	.4890E+00
.2000E-04	.4910E+00
.2100E-04	.4926E+00
.2200E-04	.4940E+00
.2300E-04	.4951E+00
.2400E-04	.4960E+00
.2500E-04	.4967E+00
.2600E-04	.4973E+00
.2700E-04	.4978E+00
.2800E-04	.4982E+00
.2900E-04	.4985E+00
.3000E-04	.4988E+00
.3100E-04	.4990E+00
.3200E-04	.4992E+00
.3300E-04	.4993E+00
.3400E-04	.4995E+00
.3500E-04	.4996E+00
.3600E-04	.4996E+00
.3700E-04	.4997E+00
.3800E-04	.4998E+00
.3900E-04	.4998E+00
.4000E-04	.4998E+00
.4100E-04	.4999E+00
.4200E-04	.4999E+00
.4300E-04	.4999E+00
.4400E-04	.4999E+00
.4500E-04	.4999E+00
.4600E-04	.5000E+00
.4700E-04	.5000E+00
.4800E-04	.5000E+00
.4900E-04	.5000E+00
.5000E-04	.5000E+00



BRANCH 2 CURRENT

TIME	VALUE
0.	.1000E-02
.1000E-05	.9091E-03
.2000E-05	.8347E-03
.3000E-05	.7739E-03
.4000E-05	.7241E-03
.5000E-05	.6833E-03
.6000E-05	.6500E-03
.7000E-05	.6227E-03
.8000E-05	.6004E-03
.9000E-05	.5822E-03
.1000E-04	.5672E-03
.1100E-04	.5550E-03
.1200E-04	.5450E-03
.1300E-04	.5368E-03
.1400E-04	.5301E-03
.1500E-04	.5246E-03
.1600E-04	.5202E-03
.1700E-04	.5165E-03
.1800E-04	.5135E-03
.1900E-04	.5110E-03
.2000E-04	.5090E-03
.2100E-04	.5074E-03
.2200E-04	.5060E-03
.2300E-04	.5049E-03
.2400E-04	.5040E-03
.2500E-04	.5033E-03
.2600E-04	.5027E-03
.2700E-04	.5022E-03
.2800E-04	.5018E-03
.2900E-04	.5015E-03
.3000E-04	.5012E-03
.3100E-04	.5010E-03
.3200E-04	.5008E-03
.3300E-04	.5007E-03
.3400E-04	.5005E-03
.3500E-04	.5004E-03
.3600E-04	.5004E-03
.3700E-04	.5003E-03
.3800E-04	.5002E-03
.3900E-04	.5002E-03
.4000E-04	.5002E-03
.4100E-04	.5001E-03
.4200E-04	.5001E-03
.4300E-04	.5001E-03
.4400E-04	.5001E-03
.4500E-04	.5001E-03
.4600E-04	.5000E-03
.4700E-04	.5000E-03
.4800E-04	.5000E-03
.4900E-04	.5000E-03
.5000E-04	.5000E-03



Sample Problem # 7:

Large time constant spread problem. - slow response.

The network is identical to that of problem # 6. However, this time, we desire the step response associated with the large time constant.

Specifications:

—initial time = 0 (TSTART = 0.0)

—final time = 500 sec (TEND = 500.0)

—time increment = 10 sec. (H = 10.0)

—desire results, at each time point, to be outputed. (NSTEP = 1)

—do not desire tree information and F to be outputed (NCONT = 0)

—desire internal automatic current scaling. (NSCALE ≠ 1)

—initial voltages are

$V_{c3}(0) = 0.5$ v

$V_{c5}(0) = 0$ v

desire the following graphical outputs

Branch 3 voltage
Branch 4 current
Branch 2 current
Branch 5 voltage
} (NGRAPH = 4)

The data cards for this problem may be seen on the following page.

The central processor time, for this problem, was 3.06 seconds.

DATA CARDS: #7

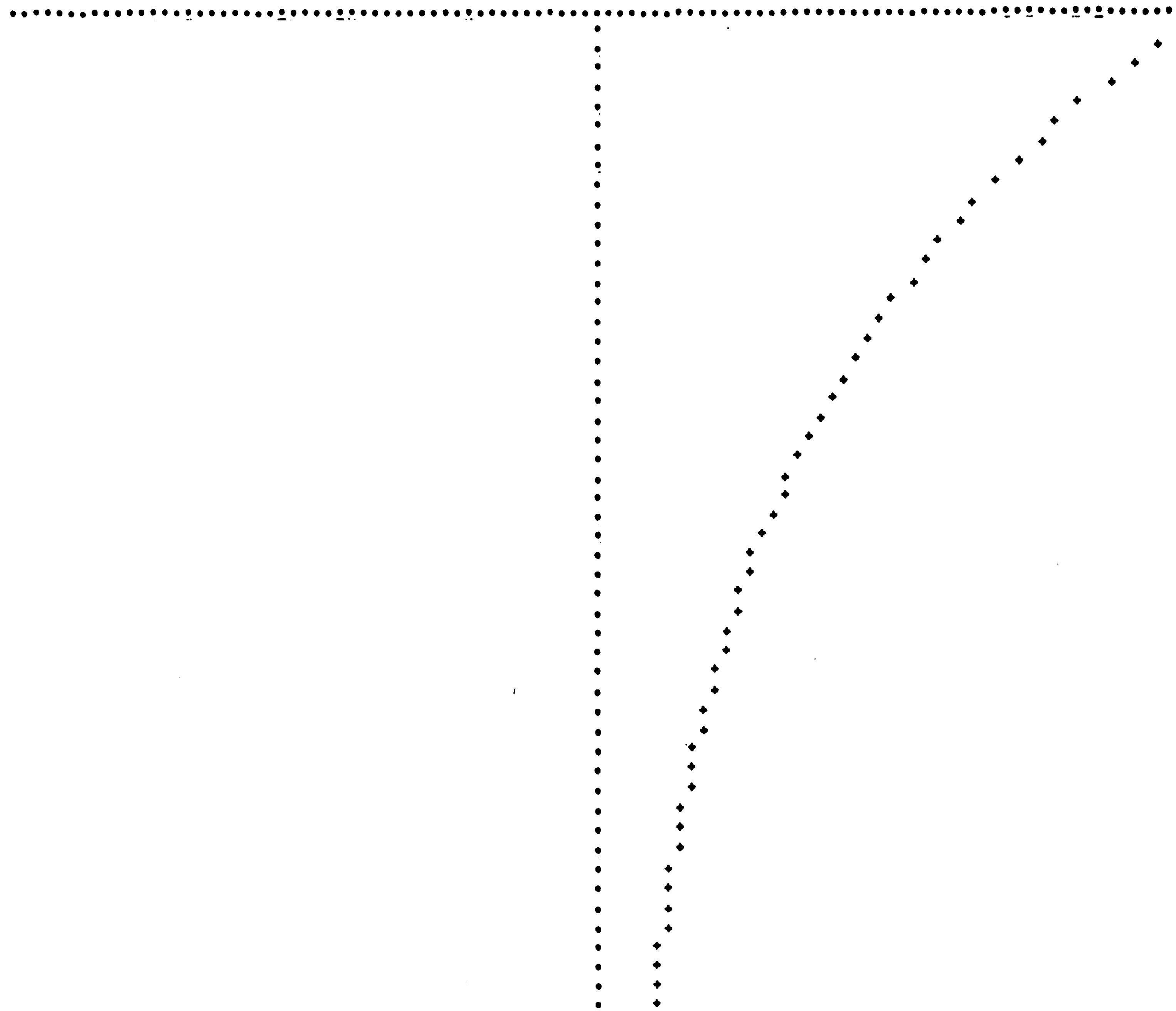
```
1234567890123456789012345678901234567890123456789012345678901234567890123456789
    4    5            0.0          500.0           10.0    1
    4
E  1C      1  4                                1.0
R  2       1  2              1000.0
C  3       2  4             1.0E-08            0.5
R  4       2  3              1000.0
C  5       3  4                 0.1
    3    0
    4    1
    2    1
    5    0
```

Comments on Sample Problem # 7:

Note that the initial voltage on C_3 has been set to 0.5 volts, which is the steady state value associated with the small time constant (see problem # 6). Failing to do the above would yield incorrect results, since the trapezoidal integration scheme, applied to branch 3, would have a step size to time constant ratio of 10^8, whereas a ratio of 0.1 is desirable for accurate integration. With the proper initial condition, C_3 becomes, effectively, a voltage source, and the integration problem does not arise.

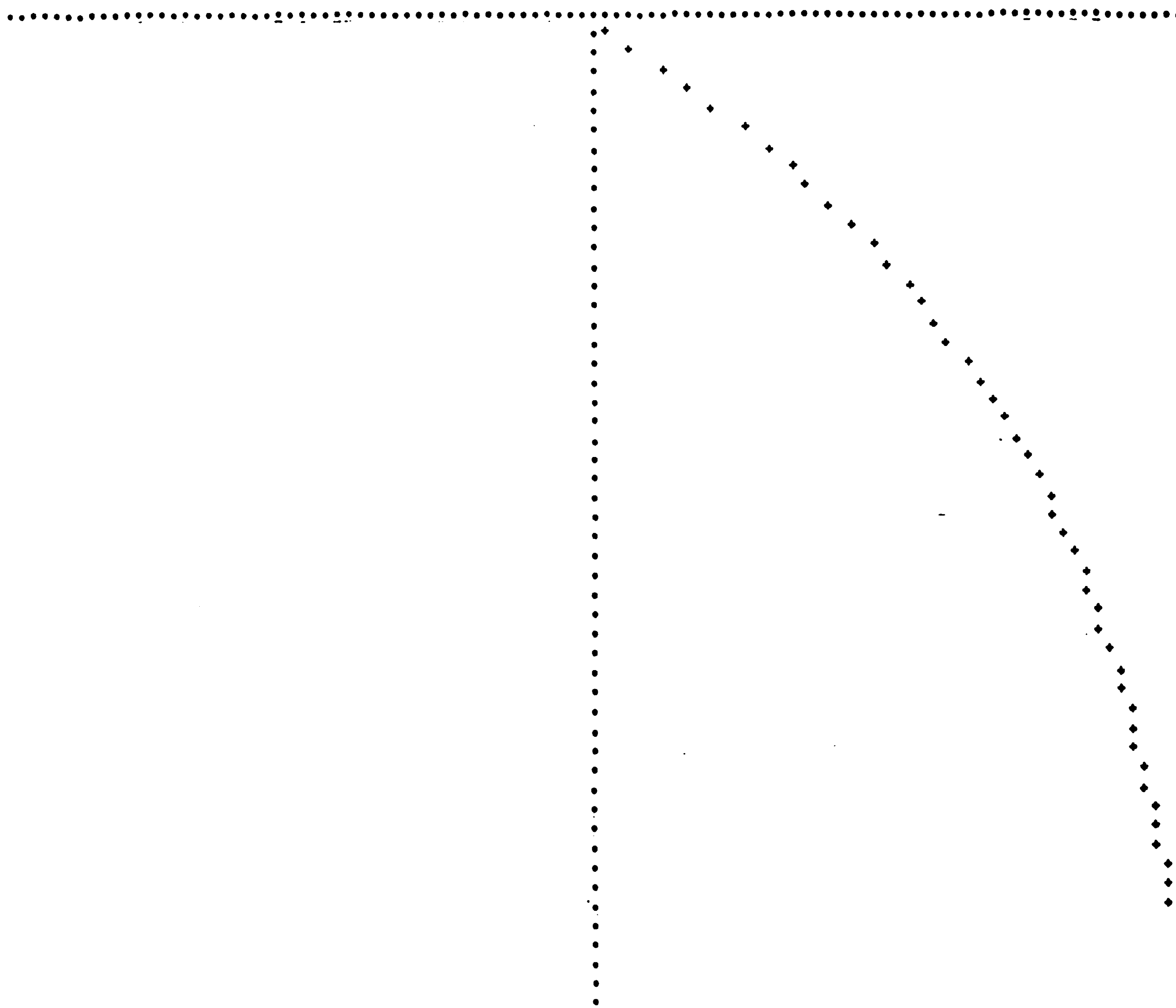
BRANCH 2 CURRENT

TIME	VALUE
0.	.5000E-03
.1000E+02	.4756E-03
.2000E+02	.4524E-03
.3000E+02	.4303E-03
.4000E+02	.4093E-03
.5000E+02	.3894E-03
.6000E+02	.3704E-03
.7000E+02	.3523E-03
.8000E+02	.3351E-03
.9000E+02	.3188E-03
.1000E+03	.3032E-03
.1100E+03	.2884E-03
.1200E+03	.2744E-03
.1300E+03	.2610E-03
.1400E+03	.2483E-03
.1500E+03	.2361E-03
.1600E+03	.2246E-03
.1700E+03	.2137E-03
.1800E+03	.2032E-03
.1900E+03	.1933E-03
.2000E+03	.1839E-03
.2100E+03	.1749E-03
.2200E+03	.1664E-03
.2300E+03	.1583E-03
.2400E+03	.1506E-03
.2500E+03	.1432E-03
.2600E+03	.1362E-03
.2700E+03	.1296E-03
.2800E+03	.1233E-03
.2900E+03	.1172E-03
.3000E+03	.1115E-03
.3100E+03	.1061E-03
.3200E+03	.1009E-03
.3300E+03	.9599E-04
.3400E+03	.9131E-04
.3500E+03	.8686E-04
.3600E+03	.8262E-04
.3700E+03	.7859E-04
.3800E+03	.7475E-04
.3900E+03	.7111E-04
.4000E+03	.6764E-04
.4100E+03	.6434E-04
.4200E+03	.6120E-04
.4300E+03	.5822E-04
.4400E+03	.5538E-04
.4500E+03	.5267E-04
.4600E+03	.5011E-04
.4700E+03	.4766E-04
.4800E+03	.4534E-04
.4900E+03	.4312E-04
.5000E+03	.4102E-04



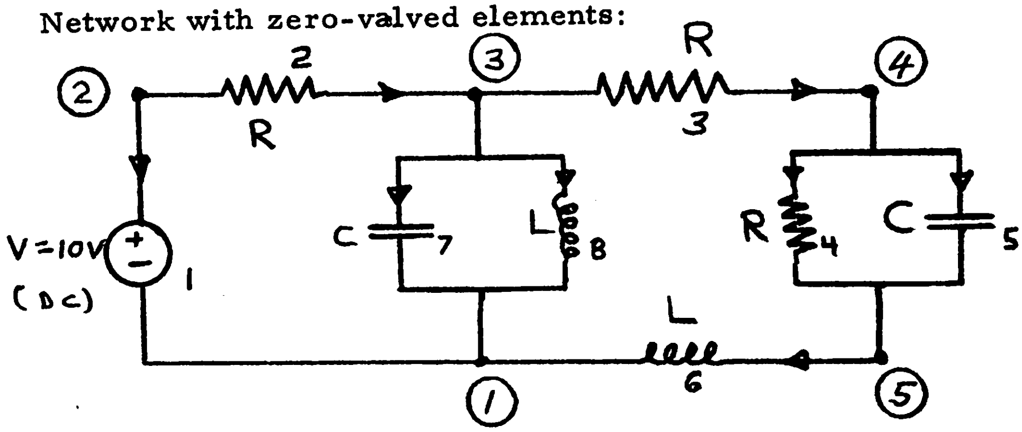
BRANCH 5 VOLTAGE

TIME	VALUE
0.	0.
.1000E+02	.4878E-01
.2000E+02	.9518E-01
.3000E+02	.1393E+00
.4000E+02	.1813E+00
.5000E+02	.2212E+00
.6000E+02	.2592E+00
.7000E+02	.2954E+00
.8000E+02	.3297E+00
.9000E+02	.3624E+00
.1000E+03	.3935E+00
.1100E+03	.4231E+00
.1200E+03	.4513E+00
.1300E+03	.4780E+00
.1400E+03	.5035E+00
.1500E+03	.5277E+00
.1600E+03	.5507E+00
.1700E+03	.5727E+00
.1800E+03	.5935E+00
.1900E+03	.6133E+00
.2000E+03	.6322E+00
.2100E+03	.6501E+00
.2200E+03	.6672E+00
.2300E+03	.6834E+00
.2400E+03	.6989E+00
.2500E+03	.7136E+00
.2600E+03	.7275E+00
.2700E+03	.7408E+00
.2800E+03	.7535E+00
.2900E+03	.7655E+00
.3000E+03	.7769E+00
.3100E+03	.7878E+00
.3200E+03	.7982E+00
.3300E+03	.8080E+00
.3400E+03	.8174E+00
.3500E+03	.8263E+00
.3600E+03	.8348E+00
.3700E+03	.8428E+00
.3800E+03	.8505E+00
.3900E+03	.8578E+00
.4000E+03	.8647E+00
.4100E+03	.8713E+00
.4200E+03	.8776E+00
.4300E+03	.8836E+00
.4400E+03	.8892E+00
.4500E+03	.8947E+00
.4600E+03	.8998E+00
.4700E+03	.9047E+00
.4800E+03	.9093E+00
.4900E+03	.9138E+00
.5000E+03	.9180E+00



Sample Problem # 8:

Network with zero-valved elements:



$R_2 = 1$ NN = 5

$R_3 = 0$ NB = 8

$R_4 = 2$

$C_5 = 0$

$L_6 = 0$

$C_7 = 1$

$L_8 = 1$

Specifications:

— initial time = 0 (TSTART = 0.0)

— final time = 5 sec. (TEND = 5.1)

— time increment = 0.1 sec. (H = 0.1)

— desire tree and F to be outputed (NCONT = 1)

— desire all voltages and currents outputed at every fifteenth time point (NSTEP = 50)

— desire internal, automatic current scaling (NSCALE ≠ 1)

— zero initial conditions

The data cards required for this problem may be seen on the following page.

The central processor time for this problem was 2.49 seconds.

Comments on Sample Problem # 8:

Note that at the initial time point, all capacitances are tree branches and all inductances are links, as desired. For the remaining time points, zero valved R's and L's are tree branches and zero valued C's are links. Such a tree will always exist as long as there are no loops of short circuits (independent voltage sources, zero valued R's and L's), and no cutsets of open circuits (independent current sources and zero valued C's). Should the above conditions not be satisified, an arithmetic error will result during the execution of the program.

DATA CARDS : PROBLEM #8

```
    5    8            0.0            5.1            0.1   50                   1
             16
E  10      2  1                               10.0
R   2      2  3                 1.0
R   3      3  4                 0.0
R   4      4  5                 2.0
C   5      4  5                 0.0
L   6      5  1                 0.0
C   7      3  1                 1.0
L   8      3  1                 1.0
    1    0
    2    0
    3    0
    4    0
    5    0
    6    0
    7    0
    8    0
    1    1
    2    1
    3    1
    4    1
    5    1
```

PROBLEM #8 CONT'D

```
    6    /
    7    /
    8    /
```

TIME = 0.

BRANCH	1	VOLTAGE	.10000E+02
BRANCH	2	VOLTAGE	.10000E+02
BRANCH	3	VOLTAGE	0.
BRANCH	4	VOLTAGE	0.
BRANCH	5	VOLTAGE	0.
BRANCH	6	VOLTAGE	0.
BRANCH	7	VOLTAGE	0.
BRANCH	8	VOLTAGE	0.
BRANCH	1	CURRENT	-.10000E+02
BRANCH	2	CURRENT	.10000E+02
BRANCH	3	CURRENT	0.
BRANCH	4	CURRENT	0.
BRANCH	5	CURRENT	0.
BRANCH	6	CURRENT	0.
BRANCH	7	CURRENT	.10000E+02
BRANCH	8	CURRENT	0.

TIME = .50000E+01

BRANCH	1	VOLTAGE	.10000E+02
BRANCH	2	VOLTAGE	.10060E+02
BRANCH	3	VOLTAGE	0.
BRANCH	4	VOLTAGE	-.59957E-01
BRANCH	5	VOLTAGE	-.59957E-01
BRANCH	6	VOLTAGE	0.
BRANCH	7	VOLTAGE	-.59957E-01
BRANCH	8	VOLTAGE	-.59957E-01
BRANCH	1	CURRENT	-.10060E+02
BRANCH	2	CURRENT	.10060E+02
BRANCH	3	CURRENT	-.29979E-01
BRANCH	4	CURRENT	-.29979E-01
BRANCH	5	CURRENT	0.
BRANCH	6	CURRENT	-.29979E-01
BRANCH	7	CURRENT	-.18740E+00
BRANCH	8	CURRENT	.10277E+02

APPENDIX D

CANINE Fortran IV Listing (CDC 6400)

```
      PROGRAM CANINE(INPUT,OUTPUT)
      INTEGER CNTSOR
      INTEGER TEMP
      INTEGER TYPE
      INTEGER SORTYPE,CONTYPE
      INTEGER CONTTEM,SORTEM
      COMMON /BLOCK1/ NNP(200),NP(200)
      COMMON /BLOCK2/ IBRAN(200),LEAV(200),LENT(200)
      COMMON /BLOCK3/ TYPE(200),VALUE(200),F(49,151),NOEL(7)
      COMMON /BLOCK4/ ITBRAN(200),LEAVT(200),LENTT(200)
      COMMON /BLOCK5/ IOUT(200),NOUT(200),ITEST(200)
      COMMON /BLOCK6/ TEMP(7),E(200),GRAD(200),CONST(200)
      COMMON /BLOCK7/ IDIMEN,ALPHA,FUNCT,NIC,NITER,H,EPS,NN,NB
      COMMON /BLOCK8/ V(200,2),C(200,2)
      COMMON /BLOCK9/ SORTYPE(200),CONTYPE(200),KONBRAN(200)
      COMMON /BLOCK10/ COND(200),IDEL,TSTART,TEND,NSTEP
      COMMON /BLOCK11/ SORVAL(5,100),TIMEPT(5,100),SNSOID(5,3),NNI(5)
      COMMON /BLOCK12/ CONTTEM(200),SORTEM(200),CONDTEM(200),KONTEM(200)
      COMMON /BK13/ ISTEP,OLDVAL(5),SECVAL(5),OLDTIME(5),SECTIME(5),NNJ(
     15)
      COMMON /BLOCK14/ KP(5)
      COMMON /BLOCK15/ NIT,JOUT,NGRAPH,NALLOUT
      COMMON /BLOCK16/ GRAF(200,5),JGRAPH(5),IGRAPH(5),NPRINT,SCALE,NITT
      COMMON /BLOCK18/ VAROUT(200)
      COMMON /BLOCK19/ CNTSOR(200)
      COMMON /BLOCK20/ NCONT,KLOOP,KPP
      COMMON /BLOCK21/ NST(5)
      COMMON /BLOCK22/ CK(200)
      COMMON /BLOCK24/ SCAL,NSCALE
      COMMON /BLOCK25/ X(200),Y(200),Z(200,2)
      DIMENSION HH(20,20)
      NDATA=0
      NLNEAR=0
1     NIT=0
      NPRINT=0
      NIC=0
      IF (NDATA.EQ.0) GO TO 2
      CALL SECOND (T)
      TO=T-TU
      PRINT 13, TO
2     CALL SECOND (TU)
      CALL READIN
      IDEL=NSTEP-1
      NDATA=NDATA+1
      CALL KONST
      CALL INCREM
      GO TO 9
3     NIC=NIC+1
      CALL INCREM
      IF (NOEL(7).EQ.0) GO TO 5
      NK=NB-NOEL(7)+1
      DO 4 I=NK,NB
      IF (SORTYPE(I).EQ.1HC) GO TO 4
      C(I,2)=C(I,2)*SCALE
4     CONTINUE
```

```
5       NIT=0
        IF (NLNEAR.NE.0) GO TO 6
        IF (NIC.NE.1) GO TO 11
6       CALL PT
        IF (KPP.EQ.0) GO TO 7
        GO TO 8
7       SCAL=SCALE
        GO TO 11
8       CALL FCSM
        CALL KONST
9       IF (NIC.EQ.0) GO TO 11
        DO 10 I=NN,NB
        C(I,2)=C(I,2)*SCALE/SCAL
        C(I,1)=C(I,1)*SCALE/SCAL
10      CONTINUE
11      CALL MATSET (HH,NB,NN,NIC)
        CALL CALCAL
        NIT=NIT+1
        IDEL=IDEL+1
        IF (IDEL.NE.NSTEP) GO TO 12
        NPRINT=NPRINT+1
        IF (NGRAPH.GE.1) CALL GRAPH
        IF (NALLOUT.GE.1) CALL ALLOUT
        IF (JOUT.GE.1) CALL READOUT
        IDEL=0
        IF (NGRAPH.EQ.201) GO TO 1
12      IF (NIC-NITER) 3,1,1
C
13      FORMAT (1H1,20X,*THE CENTRAL PROCESSOR TIME FOR THIS PROBLEM  = *,
       1E15.5,*  SECONDS*)
        END
```

```
      SUBROUTINE READIN
C     NB IS THE NUMBER OF BRANCHES
C     NN IS THE NUMBER OF NODES
C
      INTEGER TYPE
      INTEGER TEMP
      INTEGER SORTYPE,CONTYPE
      INTEGER CONTTEM,SORTEM
      INTEGER TYTEMP
      COMMON /BLOCK1/ NNP(200),NP(200)
      COMMON /BLOCK2/ IBRAN(200),LEAV(200),LENT(200)
      COMMON /BLOCK3/ TYPE(200),VALUE(200),F(49,151),NOEL(7)
      COMMON /BLOCK4/ ITBRAN(200),LEAVT(200),LENTT(200)
      COMMON /BLOCK5/ IOUT(200),NOUT(200),ITEST(200)
      COMMON /BLOCK6/ TEMP(7),ITBB(200),TYTEMP(200),VALTEM(200)
      COMMON /BLOCK7/ IDIMEN,ALPHA,FUNCT,NIC,NITER,H,EPS,NN,NB
      COMMON /BLOCK9/ SORTYPE(200),CONTYPE(200),KONBRAN(200)
      COMMON /BLOCK10/ COND(200),IDEL,TSTART,TEND,NSTEP
      COMMON /BLOCK11/ SORVAL(5,100),TIMEPT(5,100),SNSOID(5,3),NNI(5)
      COMMON /BLOCK12/ CONTTEM(200),SORTEM(200),CONDTEM(200),KONTEM(200)
      COMMON /BK13/ ISTEP,OLDVAL(5),SECVAL(5),OLDTIME(5),SECTIME(5),NNJ(
     15)
      COMMON /BLOCK15/ NIT,JOUT,NGRAPH,NALLOUT
      COMMON /BLOCK16/ GRAF(200,5),JGRAPH(5),IGRAPH(5),NPRINT,SCALE,NITT
      COMMON /BLOCK20/ NCONT,KLOOP
      COMMON /BLOCK24/ SCAL,NSCALE
1     ISTEP=0
      IJK1=0
      IJK2=0
      KSOR=0
      KCURR=6
      READ 19, NN,NB,TSTART,TEND,H,NSTEP,EPS,NCONT
      IF (NN.EQ.0) STOP
      READ 20, NGRAPH,NALLOUT,JOUT,SCALE,NITT,NSCALE
      PRINT 21, TSTART,TEND,H
      PRINT 22, NN,NB,SCALE,NSCALE
      PRINT 23, NSTEP
      IF (H.EQ.0.0) GO TO 2
      NITER=(TEND-TSTART)/H
      GO TO 3
2     NITER=0
3     DO 9 I=1,NB
      READ 24, TYPE(I),IBRAN(I),SORTYPE(I),CONTYPE(I),KONBRAN(I),LEAV(I)
     1,LENT(I),NCARDS,VALUE(I),COND(I)
      IF (NCARDS.EQ.0) GO TO 9
      IF (TYPE(I).EQ.1HJ) GO TO 4
      IF (SORTYPE(I).EQ.1HS.OR.SORTYPE(I).EQ.1HE) GO TO 5
      IF (SORTYPE(I).EQ.1HP.OR.SORTYPE(I).EQ.1HT) GO TO 6
4     ISTEP=ISTEP+1
      IF (SORTYPE(I).EQ.1HP.OR.SORTYPE(I).EQ.1HT) GO TO 8
      IF (SORTYPE(I).EQ.1HS.OR.SORTYPE(I).EQ.1HE) GO TO 7
      GO TO 9
5     KSOR=KSOR+1
      NNJ(KSOR)=0
      READ 25, (SNSOID(KSOR,J),J=1,3)
```

```
      GO TO 9
6     KSOR=KSOR+1
      NNJ(KSOR)=0
      READ 26, NNI(KSOR)
      NNN=NNI(KSOR)
      READ 27, (SORVAL(KSOR,J),J=1,NNN)
      READ 27, (TIMEPT(KSOR,J),J=1,NNN)
      GO TO 9
7     KCURR=KCURR-1
      NNJ(KCURR)=0
      READ 25, (SNSOID(KCURR,J),J=1,3)
      GO TO 9
8     KCURR=KCURR-1
      NNJ(KCURR)=0
      READ 26, NNI(KCURR)
      NNN=NNI(KCURR)
      READ 27, (SORVAL(KCURR,J),J=1,NNN)
      READ 27, (TIMEPT(KCURR,J),J=1,NNN)
9     CONTINUE
      IF (NGRAPH.EQ.0) GO TO 10
      READ 28, ((IGRAPH(I),JGRAPH(I)),I=1,NGRAPH)
10    IF (NALLOUT.GE.1.OR.NGRAPH.GE.1) GO TO 11
      READ 28, ((IOUT(I),ITEST(I)),I=1,JOUT)
11    PRINT 18
      PRINT 29
      DO 12 I=1,NB
      PRINT 30, IBRAN(I),TYPE(I),VALUE(I),LEAV(I),LENT(I),COND(I)
12    CONTINUE
      DO 14 I=1,NB
      IF (SORTYPE(I).NE.1H ) GO TO 13
      GO TO 14
13    IF (IJK1.EQ.0) PRINT 31
      IJK1=1
      A=7HVOLTAGE
      IF (TYPE(I).EQ.1HJ) A=7HCURRENT
      PRINT 32, A,IBRAN(I),SORTYPE(I)
14    CONTINUE
      DO 17 I=1,NB
      IF (CONTYPE(I).NE.1H ) GO TO 15
      GO TO 17
15    IF (IJK2.EQ.0) PRINT 33
      IJK2=1
      IF (TYPE(I).EQ.1HI) GO TO 16
      A=7HCURRENT
      IF (CONTYPE(I).EQ.1HV) A=7HVOLTAGE
      PRINT 34, A,IBRAN(I),KONBRAN(I)
      GO TO 17
16    A=7HCURRENT
      IF (CONTYPE(I).EQ.1HV) A=7HVOLTAGE
      PRINT 35, A,IBRAN(I),KONBRAN(I)
17    CONTINUE
      IDIMEN=NB-NN+1
      KLOOP=0
      CALL PT
      IF (KLOOP.EQ.1) GO TO 1
```

```
      CALL FCSM
      RETURN
C
18    FORMAT (1H1,55X,*THIS IS THE GIVEN NETWORK *)
19    FORMAT (I5,I5,E15.3,E15.3,E15.3,I5,E15.3,I5)
20    FORMAT (3I5,E15.5,I5,I5)
21    FORMAT (1H1,/////                                          ,40X,* STARTI
     1NG TIME = *,E15.5,*   END TIME = *,E15.5,///,40X,*STEP SIZE = *,
     2 E15.5)
22    FORMAT (1H0,///,30X,*NUMBER OF NODES = *,I5,*    NUMBER OF BRANCHE
     1S = *,I5,///,30X,*SCALE FACTOR = *,E15.5,///,30X,
     2                                               *NSCALE = *,I5)
23    FORMAT (1H0,///,30X,* NUMBER OF TIME ITERATIONS PER OUTPUT = *,I5)
24    FORMAT (A1,I3,A1,A1,I3,1X,I2,1X,I2,1X,I2,2X,E15.4,E15.4)
25    FORMAT ( 3E10.3)
26    FORMAT ( I5 )
27    FORMAT ( 8E10.3 )
28    FORMAT (2I5)
29    FORMAT (1H0,///,55X,*UNITS ARE OHMS, FARADS AND HENRYS *)
30    FORMAT (1H0, 3X,*BRANCH NUMBER *,2X,I3,4X,* IS A      *,A1,*     OF
     1 VALUE  *,E12.5,*  LEAVING NODE   *,I3,*  AND ENTERING NODE  *,I3,
     22X,* COND = *,E10.3)
31    FORMAT (1H1,////,30X,*INDEPENDENT SOURCES*,///)
32    FORMAT (1H0,10X,A7,  * SOURCE BRANCH *,I3,* IS OF TYPE  *,A1)
33    FORMAT (1H-,///,30X,*CONTROLLED SOURCES *,///)
34    FORMAT (1H0,10X,A7,* CONTROLLED VOLTAGE SOURCE*,I3,* IS CONTROLLED
     1 BY BRANCH*,I5)
35    FORMAT (1H0,10X,A7,* CONTROLLED CURRENT SOURCE*,I3,* IS CONTROLLED
     1 BY BRANCH*,I5)
      END
```

```
      SUBROUTINE PT
      INTEGER TEMP01,TEMP02,TEMP03,TEMP04,TEMP06,TEMP07,TEMP09
      INTEGER TYTEMP
      INTEGER TYPE
      INTEGER TEMP
      INTEGER CONTYPE,SORTYPE
      INTEGER SORTEM,CONTTEM
      INTEGER CNTSOR
      COMMON /BLOCK1/ NNP(200),NP(200)
      COMMON /BLOCK2/ IBRAN(200),LEAV(200),LENT(200)
      COMMON /BLOCK3/ TYPE(200),VALUE(200),F(49,151),NOEL(7)
      COMMON /BLOCK4/ ITBRAN(200),LEAVT(200),LENTT(200)
      COMMON /BLOCK5/ IOUT(200),NOUT(200),ITEST(200)
      COMMON /BLOCK6/ TEMP(7),ITBB(200),TYTEMP(200),VALTEM(200)
      COMMON /BLOCK7/ IDIMEN,ALPHA,FUNCT,NIC,NITER,H,EPS,NN,NB
      COMMON /BLOCK8/ V(200,2),C(200,2)
      COMMON /BLOCK9/ SORTYPE(200),CONTYPE(200),KONBRAN(200)
      COMMON /BLOCK10/ COND(200),IDEL,TSTART,TEND,NSTEP
      COMMON /BLOCK12/ CONTTEM(200),SORTEM(200),CONDTEM(200),KONTEM(200)
      COMMON /BLOCK15/ NIT,JOUT,NGRAPH,NALLOUT
      COMMON /BLOCK16/ GRAF(200,5),JGRAPH(5),IGRAPH(5),NPRINT,SCALE,NITT
      COMMON /BLOCK18/ CONST(200)
      COMMON /BLOCK19/ CNTSOR(200)
      COMMON /BLOCK20/ NCONT,KLOOP,KPP
      COMMON /BLOCK21/ NST(5)
      KK=NN-1
      KM=0
      KKTT=0
      IF (NIC.NE.0) GO TO 5
      TEMP(1)=1HE
      TEMP(2)=1HV
      TEMP(3)=1HC
      TEMP(4)=1HR
      TEMP(5)=1HL
      TEMP(6)=1HI
      TEMP(7)=1HJ
      DO 3 K=1,7
      KT=0
      DO 2 I=1,NB
      IF (TYPE(I).EQ.TEMP(K)) GO TO 1
      GO TO 2
1     KM=KM+1
      ITBRAN(KM)=I
      TYTEMP(KM)=TEMP(K)
      VALTEM(KM)=VALUE(I)
      KT=KT+1
      CONTTEM(KM)=CONTYPE(I)
      SORTEM(KM)=SORTYPE(I)
      CONDTEM(KM)=COND(I)
      KONTEM(KM)=KONBRAN(I)
2     CONTINUE
      NOEL(K)=KT
3     CONTINUE
      DO 4 I=1,NB
      K=ITBRAN(I)
```

```
      LEAVT(I)=LEAV(K)
      LENTT(I)=LENT(K)
      TYPE(I)=TYTEMP(I)
      VALUE(I)=VALTEM(I)
      CONTYPE(I)=CONTTEM(I)
      SORTYPE(I)=SORTEM(I)
      COND(I)=CONDTEM(I)
      KONBRAN(I)=KONTEM(I)
4     CONTINUE
      IF (NOEL(4).EQ.0) GO TO 15
      N=NOEL(1)+NOEL(2)+NOEL(3)+1
      MM=N+NOEL(4)-1
      GO TO 6
5     CONTINUE
      MM=NB-NOEL(7)
      N=NOEL(1)+1
6     M=MM-1
      DO 7 I=N,MM
      IF (TYPE(I).EQ.1HV) CONST(I)=1.0E-50
      IF (TYPE(I).EQ.1HC) GO TO 8
      IF (TYPE(I).EQ.1HR) CONST(I)=VALUE(I)
      IF (TYPE(I).EQ.1HL) CONST(I)=2.0*VALUE(I)/H
      IF (TYPE(I).EQ.1HI) CONST(I)=1.0E+50
7     CONTINUE
      GO TO 9
8     IF (VALUE(I).EQ.0.0) CONST(I)=1.0E+51
      IF (VALUE(I).NE.0.0) CONST(I)=H/(2.0*VALUE(I))
      GO TO 7
9     KPP=0
      IF (N.GT.M) GO TO 15
      DO 14 I=N,M
      KP=0
      AMIN=CONST(I)
      K=I+1
      DO 11 J=K,MM
      IF (CONST(J).LT.AMIN) GO TO 10
      GO TO 11
10    AMIN=CONST(J)
      NP(I)=J
      KP=1
      KPP=1
11    CONTINUE
      IF (KP.EQ.0) GO TO 14
      J=NP(I)
      TEMP01=ITBRAN(I)
      TEMP02=LEAVT(I)
      TEMP03=LENTT(I)
      TEMP04=TYPE(I)
      TEMP05=VALUE(I)
      TEMP06=CONTYPE(I)
      TEMP07=SORTYPE(I)
      TEMP08=COND(I)
      TEMP09=KONBRAN(I)
      TEMP10=CONST(I)
      IF (NIC.EQ.0) GO TO 12
```

```
      TEMP11=V(I,2)
      TEMP12=C(I,2)
      TEMP13=V(I,1)
      TEMP14=C(I,1)
12    ITBRAN(I)=ITBRAN(J)
      LEAVT(I)=LEAVT(J)
      LENTT(I)=LENTT(J)
      TYPE(I)=TYPE(J)
      VALUE(I)=VALUE(J)
      CONTYPE(I)=CONTYPE(J)
      SORTYPE(I)=SORTYPE(J)
      COND(I)=COND(J)
      KONBRAN(I)=KONBRAN(J)
      CONST(I)=CONST(J)
      IF (NIC.EQ.0) GO TO 13
      V(I,2)=V(J,2)
      C(I,2)=C(J,2)
      V(I,1)=V(J,1)
      C(I,1)=C(J,1)
13    ITBRAN(J)=TEMP01
      LEAVT(J)=TEMP02
      LENTT(J)=TEMP03
      TYPE(J)=TEMP04
      VALUE(J)=TEMP05
      CONTYPE(J)=TEMP06
      SORTYPE(J)=TEMP07
      COND(J)=TEMP08
      KONBRAN(J)=TEMP09
      CONST(J)=TEMP10
      IF (NIC.EQ.0) GO TO 14
      V(J,2)=TEMP11
      C(J,2)=TEMP12
      V(J,1)=TEMP13
      C(J,1)=TEMP14
14    CONTINUE
15    IF (2.GT.KK) GO TO 32
      DO 25 I=2,KK
16    NE=LENTT(I)
      NL=LEAVT(I)
      NP(1)=I
      NNP(1)=1
      M=I-1
      MT=0
      KT=1
17    DO 19 JJJ=1,M
      J=M+1-JJJ
      IF (MT.EQ.J) GO TO 18
      IF (LEAVT(J).EQ.NE) GO TO 23
      IF (LENTT(J).EQ.NE) GO TO 24
18    CONTINUE
19    CONTINUE
20    IF (KT.EQ.1) GO TO 25
      KA=NNP(KT)
      KB=NP(KT)
      M=KB-1
```

```
      IF (M.EQ.0) GO TO 21
      MT=NP(KT-1)
      KT=KT-1
      IF (KA.EQ.-1) GO TO 22
      NE=LEAVT(KB)
      GO TO 17
21    KT=KT-1
      GO TO 20
22    NE=LENTT(KB)
      GO TO 17
23    IF (LENTT(J).EQ.NL) GO TO 26
      NE=LENTT(J)
      KT=KT+1
      NP(KT)=J
      NNP(KT)=1
      MT=J
      M=I-1
      GO TO 17
24    IF (LEAVT(J).EQ.NL) GO TO 26
      NE=LEAVT(J)
      KT=KT+1
      NP(KT)=J
      NNP(KT)=-1
      MT=J
      M=I-1
      GO TO 17
25    CONTINUE
      GO TO 32
26    LAE=NOEL(1)
      IF (I.LE.LAE) GO TO 52
27    TEMPO1=ITBRAN(I)
      TEMPO2=LEAVT(I)
      TEMPO3=LENTT(I)
      TEMPO4=TYPE(I)
      TEMPO5=VALUE(I)
      TEMPO6=CONTYPE(I)
      TEMPO7=SORTYPE(I)
      TEMPO8=COND(I)
      TEMPO9=KONBRAN(I)
      IF (NIC.EQ.0) GO TO 28
      TEMP11=V(I,2)
      TEMP12=C(I,2)
      TEMP13=V(I,1)
      TEMP14=C(I,1)
28    CONTINUE
      NLB=NB-1
      IF (NIC.NE.0) NLB=NB-NOEL(7)-1
      DO 30 JN=I,NLB
      JF=JN+1
      ITBRAN(JN)=ITBRAN(JF)
      LEAVT(JN)=LEAVT(JF)
      LENTT(JN)=LENTT(JF)
      TYPE(JN)=TYPE(JF)
      VALUE(JN)=VALUE(JF)
      CONTYPE(JN)=CONTYPE(JF)
```

```
      SORTYPE(JN)=SORTYPE(JE)
      COND(JN)=COND(JE)
      KONBRAN(JN)=KONBRAN(JE)
      IF (NIC.EQ.0) GO TO 29
      V(JN,2)=V(JE,2)
      C(JN,2)=C(JE,2)
      V(JN,1)=V(JE,1)
      C(JN,1)=C(JE,1)
29    CONTINUE
30    CONTINUE
      NZ=NB
      IF (NIC.NE.0) NZ=NB-NOEL(7)
      ITBRAN(NZ)=TEMP01
      LEAVT(NZ)=TEMP02
      LENTT(NZ)=TEMP03
      TYPE(NZ)=TEMP04
      VALUE(NZ)=TEMP05
      CONTYPE(NZ)=TEMP06
      SORTYPE(NZ)=TEMP07
      COND(NZ)=TEMP08
      KONBRAN(NZ)=TEMP09
      IF (NIC.EQ.0) GO TO 31
      V(NZ,2)=TEMP11
      C(NZ,2)=TEMP12
      V(NZ,1)=TEMP13
      C(NZ,1)=TEMP14
31    CONTINUE
      IF (KKTT.EQ.1) GO TO 35
      GO TO 16
32    CONTINUE
      DO 33 L=1,KK
      IF (TYPE(L).EQ.TEMP(7)) GO TO 53
33    CONTINUE
      IF (NIC.NE.0) GO TO 36
      DO 35 I=NN,NB
      IF (TYPE(I).EQ.TEMP(7)) GO TO 34
      GO TO 35
34    KKTT=1
      GO TO 27
35    CONTINUE
36    DO 37 I=1,NB
      K=ITBRAN(I)
      NP(I)=IBRAN(K)
37    CONTINUE
      DO 38 I=1,NB
      CNTSOR(I)=0
38    CONTINUE
      IF (JOUT.EQ.0) GO TO 41
      DO 40 I=1,JOUT
      K=0
39    K=K+1
      IF (IOUT(I).NE.NP(K)) GO TO 39
      NOUT(I)=K
40    CONTINUE
41    IF (NGRAPH.EQ.0) GO TO 44
```

```
      DO 43 I=1,NGRAPH
      K=0
42    K=K+1
      IF (IGRAPH(I).NE.NP(K)) GO TO 42
      NST(I)=K
43    CONTINUE
44    N=NOEL(1)+1
      M=NB-NOEL(7)
      DO 50 I=N,M
      IF (TYPE(I).EQ.1HI) GO TO 45
      IF (TYPE(I).EQ.1HV) GO TO 45
      GO TO 50
45    K=KONBRAN(I)
      L=0
46    L=L+1
      IF (L.GT.NOEL(1)) GO TO 47
      IF (NP(L).NE.K) GO TO 46
      GO TO 49
47    L=NB-NOEL(7)
48    L=L+1
      IF (NP(L).NE.K) GO TO 48
49    CNTSOR(I)=L
      CNTSOR(L)=I
50    CONTINUE
      IF (NCONT.EQ.0) GO TO 54
      PRINT 55
      PRINT 56, ((I,NP(I)),I=1,KK)
      PRINT 57, ((I,NP(I)),I=NN,NB)
      PRINT 58
      DO 51 I=1,NB
      PRINT 59, NP(I),TYPE(I),VALUE(I),LEAVT(I),LENTT(I),COND(I)
51    CONTINUE
      RETURN
52    PRINT 60
      KLOOP=1
      RETURN
53    PRINT 61
      KLOOP=1
54    RETURN
C
55    FORMAT (1H1,20X,*CORRESPONDENCE BETWEEN ORIGINAL TOPOLOGY AND NEW
     1TOPOLOGY*,///)
56    FORMAT (1H0,30X,*TREE BRANCH*,I4,5X,*CORRESPONDS TO BRANCH *,I4)
57    FORMAT (1H0,33X,* LINK *,2X,I4,5X,*CORRESPONDS TO BRANCH *,I4)
58    FORMAT (1H1)
59    FORMAT (1H0, 3X,*BRANCH NUMBER *,2X,I3,4X,* IS A      *,A1,*      OF
     1 VALUE  *,F12.5,*  LEAVING NODE   *,I3,*  AND ENTERING NODE  *,I3,
     22X,* COND = *,F10.3)
60    FORMAT (1H0,30X,*THERE IS A VOLTAGE LOOP IN NETWORK*)
61    FORMAT (1H0,30X,*THERE IS A CURRENT SOURCE CUTSET IN NETWORK*)
      END
```

```
      SUBROUTINE FCSM
      INTEGER TEMP
      INTEGER TYPE
      INTEGER TYTEMP
      COMMON /BLOCK1/ NNP(200),NP(200)
      COMMON /BLOCK3/ TYPE(200),VALUE(200),F(49,151),NOEL(7)
      COMMON /BLOCK4/ ITBRAN(200),LEAVT(200),LENTT(200)
      COMMON /BLOCK5/ IOUT(10),ITEST(10),NOUT(10)
      COMMON /BLOCK6/ TEMP(7),ITBB(200),TYTEMP(200),VALTEM(200)
      COMMON /BLOCK7/ IDIMEN,ALPHA,FUNCT,NIC,NITER,H,EPS,NN,NB
      COMMON /BLOCK20/ NCONT,KLOOP
C     SET MATRIX TO ZERO
      KK=NN-1
      NLB=NB-NN+1
      DO 1 I=1,NLB
      DO 1 J=1,KK
      F(J,I)=0.0
1     CONTINUE
      DO 8 I=NN,NB
      NE=LENTT(I)
      NL=LEAVT(I)
      IO=I-NN+1
C     NP STORES THE PREVIOUS TREE BRANCH
C     NNP STORES ITS DIRECTION
      NP(1)=I
      NNP(1)=1
      M=1
      KT=1
2     DO 3 J=M,KK
      IF (J.EQ.NP(KT)) GO TO 3
      IF (LEAVT(J).EQ.NE) GO TO 6
      IF (LENTT(J).EQ.NE) GO TO 7
3     CONTINUE
4     M=NP(KT)+1
      KA=NNP(KT)
      KB=NP(KT)
      F(KB,IO)=0.0
      KT=KT-1
      IF (M.GE.NN) GO TO 4
      IF (KA.EQ.1) GO TO 5
      NE=LENTT(KB)
      GO TO 2
5     NE=LEAVT(KB)
      GO TO 2
6     F(J,IO)=-1.0
      KT=KT+1
      IF (LENTT(J).EQ.NL) GO TO 8
      NE=LENTT(J)
      NP(KT)=J
      NNP(KT)=1
      M=1
      GO TO 2
7     F(J,IO)=1.0
      KT=KT+1
      IF (LEAVT(J).EQ.NL) GO TO 8
```

```
      NE=LEAVT(J)
      NP(KT)=J
      NNP(KT)=-1
      M=1
      GO TO 2
8     CONTINUE
      IF (NCONT.EQ.0) GO TO 16
      NLB=NB-NN+1
      KK=NN-1
      NCOUNT=-1
      NT=NLB
      NPAGE=1
      PRINT 17, NPAGE
      PRINT 18
      PRINT 19, (J,J=1,NLB)
9     NT=NT-25
      NCOUNT=NCOUNT+1
      NS1=1+25*NCOUNT
      NS2=25+25*NCOUNT
      IF (NT.LE.0) GO TO 11
      DO 10 I=1,KK
      PRINT 20, I,(F(I,J),J=NS1,NS2)
10    CONTINUE
      NPAGE=NPAGE+1
      GO TO 9
11    IF (NT.EQ.0) GO TO 12
      GO TO 13
12    NS3=1
      NS4=25
      GO TO 14
13    NT=NT+25
      NS3=NS1
      NS4=NS1+NT-1
14    CONTINUE
      DO 15 I=1,KK
      PRINT 20, I,(F(I,J),J=NS3,NS4)
15    CONTINUE
16    RETURN
C
17    FORMAT (1H1,40X,*FUNDAMENTAL CUT SET MATRIX *,5X,*PAGE *,I3)
18    FORMAT (///)
19    FORMAT (1H0,7X,25(2X,I2,1X))
20    FORMAT (1H0,I3,3X,25(2X,F3.0))
      END
```

```
      SUBROUTINE KONST
C     CALCULATION OF CONSTANTS RELATED TO ELEMENT VALUES FOR USE IN BRAN
C     RELATIONS AND GRADIENT
      INTEGER TEMP
      INTEGER TYPE
      INTEGER SORTYPE,CONTYPE
      COMMON /BLOCK3/ TYPE(200),VALUE(200),E(49,151),NOEL(7)
      COMMON /BLOCK6/ TEMP(7),F(200),GRAD(200),CONST(200)
      COMMON /BLOCK7/ IDIMEN,ALPHA,FUNCT,NIC,NITER,H,EPS,NN,NB
      COMMON /BLOCK8/ V(200,2),C(200,2)
      COMMON /BLOCK9/ SORTYPE(200),CONTYPE(200),KOMBRAN(200)
      COMMON /BLOCK10/ COND(200),IDEL,TSTART,TEND,NSTEP
      COMMON /BLOCK11/ SORVAL(5,100),TIMEPT(5,100),SNFOID(5,3),NNI(5)
      COMMON /BK13/ ISTEP,OLDVAL(5),SECVAL(5),OLDTIME(5),SECTIME(5),NNJ(
     15)
      COMMON /BLOCK14/ NP(5)
      COMMON /BLOCK16/ GRAF(200,5),JGRAPH(5),IGRAPH(5),NPRINT,SCALE,NITT
      COMMON /BLOCK24/ SCAL,NSCALE
      IF (NIC.NE.0) GO TO 2
      DO 1 I=1,5
      SECTIME(I)=TIMEPT(I,2)
      OLDTIME(I)=TIMEPT(I,1)
      SECVAL(I)=SORVAL(I,2)
      OLDVAL(I)=SORVAL(I,1)
      NP(I)=2
    1 CONTINUE
    2 SCAL=SCALE
      IF (NSCALE.EQ.1) GO TO 5
      KW=0
      DO 4 I=NN,NB
      IF (TYPE(I).EQ.1HR) GO TO 3
      GO TO 4
    3 IF (KW.EQ.0) SCALE=VALUE(I)
      KW=1
      IF (VALUE(I).LT.SCALE) SCALE=VALUE(I)
    4 CONTINUE
      IF (KW.EQ.1) GO TO 5
      DO 135 I=1,NN
      IF (TYPE(I).EQ.1HR) GO TO 136
      GO TO 135
  136 IF (KW.EQ.0) SCALE=VALUE(I)
      KW=1
      IF (VALUE(I).GT.SCALE) SCALE=VALUE(I)
  135 CONTINUE
      IF (KW.EQ.0) SCALE=1.0
      IF (SCALE.EQ.0.0) SCALE=1.0
    5 N=NN-1
      DO 12 I=1,N
      IF (TYPE(I).EQ.1HE) GO TO 6
      IF (TYPE(I).EQ.1HV) GO TO 7
      IF (TYPE(I).EQ.1HC) GO TO 8
      IF (TYPE(I).EQ.1HR) GO TO 9
      IF (TYPE(I).EQ.1HL) GO TO 10
      IF (TYPE(I).EQ.1HI) GO TO 11
      GO TO 12
```

```
6       IF (NIC.NE.0) GO TO 12
        V(I,2)=COND(I)
        CONST(I)=0.0
        GO TO 12
7       CONST(I)=VALUE(I)
        IF (CONTYPE(I).EQ.1HI) CONST(I)=CONST(I)/SCALE
        IF (NIC.NE.0) GO TO 12
        V(I,2)=COND(I)
        GO TO 12
8       CONST(I)=H/(2.0*VALUE(I)*SCALE)
        IF (NIC.NE.0) GO TO 12
        V(I,2)=COND(I)
        GO TO 12
9       CONST(I)=VALUE(I)/SCALE
        IF (NIC.NE.0) GO TO 12
        V(I,2)=COND(I)
        GO TO 12
10      CONST(I)=2.0*VALUE(I)/(H*SCALE)
        IF (NIC.NE.0) GO TO 12
        V(I,2)=0.0
        GO TO 12
11      CONST(I)=VALUE(I)
        IF (CONTYPE(I).EQ.1HV) CONST(I)=CONST(I)*SCALE
        IF (NIC.NE.0) GO TO 12
        V(I,2)=0.
12      CONTINUE
        DO 19 I=NN,NB
        IF (TYPE(I).EQ.1HJ) GO TO 13
        IF (TYPE(I).EQ.1HV) GO TO 14
        IF (TYPE(I).EQ.1HC) GO TO 15
        IF (TYPE(I).EQ.1HR) GO TO 16
        IF (TYPE(I).EQ.1HL) GO TO 17
        IF (TYPE(I).EQ.1HI) GO TO 18
        GO TO 19
13      IF (NIC.NE.0) GO TO 19
        C(I,2)=COND(I)*SCALE
        CONST(I)=0.0
        GO TO 19
14      CONST(I)=VALUE(I)
        IF (CONTYPE(I).EQ.1HI) CONST(I)=CONST(I)/SCALE
        IF (NIC.NE.0) GO TO 19
        C(I,2)=0.0
        GO TO 19
15      CONST(I)=2.0*VALUE(I)*SCALE/H
        IF (NIC.NE.0) GO TO 19
        C(I,2)=0.0
        GO TO 19
16      CONST(I)=1.0/VALUE(I)*SCALE
        IF (NIC.NE.0) GO TO 19
        C(I,2)=COND(I)*SCALE
        GO TO 19
17      CONST(I)=H*SCALE/(2.0*VALUE(I))
        IF (NIC.NE.0) GO TO 19
        C(I,2)=COND(I)*SCALE
        GO TO 19
```

```
18    CONST(I)=VALUE(I)
      IF (CONTYPE(I).EQ.1HV) CONST(I)=CONST(I)*SCALE
      IF (NIC.NE.0) GO TO 19
      C(I,2)=COND(I)*SCALE
19    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE CALCAL
      INTEGER CNTSOR
      INTEGER TYPE
      COMMON /BLOCK3/ TYPE(200),VALUE(200),F(49,151),NOEL(7)
      COMMON /BLOCK8/ V(200,2),C(200,2)
      COMMON /BLOCK7/ IDIMEN,ALPHA,FUNCT,NIC,NITER,H,EPS,NN,NB
      COMMON /BLOCK19/ CNTSOR(200)
      N=NN-1
      DO 1 I=1,N
      C(I,2)=0.0
      DO 1 J=NN,NB
      C(I,2)=C(I,2)-F(I,J-N)*C(J,2)
1     CONTINUE
      DO 2 I=NN,NB
      V(I,2)=0.0
      DO 2 J=1,N
      V(I,2)=V(I,2)+F(J,I-N)*V(J,2)
2     CONTINUE
      RETURN
      END
```

```
      SUBROUTINE INCREM
      INTEGER TEMP
      INTEGER TYPE
      INTEGER SORTYPE,CONTYPE
      COMMON /BLOCK3/ TYPE(200),VALUE(200),F(49,151),NOEL(7)
      COMMON /BLOCK6/ TEMP(7),E(200),GRAD(200),CONST(200)
      COMMON /BLOCK7/ IDIMEN,ALPHA,FUNCT,NIC,NITER,H,EPS,NN,NB
      COMMON /BLOCK8/ V(200,2),C(200,2)
      COMMON /BLOCK9/ SORTYPE(200),CONTYPE(200),KONBRAN(200)
      COMMON /BLOCK10/ COND(200),IDEL,TSTART,TEND,NSTEP
      COMMON /BLOCK11/ SORVAL(5,100),TIMEPT(5,100),SNSOID(5,3),NNI(5)
      COMMON /BK13/ ISTEP,OLDVAL(5),SECVAL(5),OLDTIME(5),SECTIME(5),NNJ(
     15)
      COMMON /BLOCK14/ NP(5)
      LOGICAL BOOL
      IF (NIC.EQ.0) GO TO 3
      BOOL=.TRUE.
      DO 1 I=1,NB
      V(I,1)=V(I,2)
1     CONTINUE
      DO 2 I=1,NB
      C(I,1)=C(I,2)
2     CONTINUE
3     TIME=TSTART+NIC*H
      M=NOEL(1)+NOEL(7)
      K=0
      DO 15 I=1,M
      J=I
      IF (I.GE.NOEL(1)+1) J=NB-I+NOEL(1)+1
      IF (J.GT.I.AND.BOOL) GO TO 4
      GO TO 5
4     BOOL=.FALSE.
      K=5-ISTEP
5     IF (SORTYPE(J).EQ.1HC) GO TO 15
      K=K+1
      IF (SORTYPE(J).EQ.1HF.OR.SORTYPE(J).EQ.1HS) GO TO 8
      IF (SORTYPE(J).EQ.1HP) GO TO 12
      IF (TIME.LE.SECTIME(K)) GO TO 6
      IF (NNI(K).LE.NP(K)) GO TO 13
      NP(K)=NP(K)+1
      NPR=NP(K)
      OLDTIME(K)=SECTIME(K)
      SECTIME(K)=TIMEPT(K,NPR)
      OLDVAL(K)=SECVAL(K)
      SECVAL(K)=SORVAL(K,NPR)
6     IF (I.GE.NOEL(1)+1) GO TO 7
      V(J,2)=OLDVAL(K)+(SECVAL(K)-OLDVAL(K))*(TIME-OLDTIME(K))/(SECTIME(
     1K)-OLDTIME(K))
      GO TO 15
7     C(J,2)=OLDVAL(K)+(SECVAL(K)-OLDVAL(K))*(TIME-OLDTIME(K))/(SECTIME(
     1K)-OLDTIME(K))
      GO TO 15
8     W=SNSOID(K,2)
      T=SNSOID(K,3)
      IF (SORTYPE(J).EQ.1HS) GO TO 10
```

```
      IF (J.GT.I) GO TO 9
      V(J,2)=SNSOID(K,1)*EXP(W*TIME-T)
      GO TO 15
9     C(J,2)=SNSOID(K,1)*EXP(W*TIME-T)
      GO TO 15
10    IF (J.GT.I) GO TO 11
      V(J,2)=SNSOID(K,1)*SIN(W*TIME-T)
      GO TO 15
11    C(J,2)=SNSOID(K,1)*SIN(W*TIME-T)
      GO TO 15
12    CALL PERIOD (K,J)
      GO TO 15
13    L=NNI(K)
      IF (J.GT.I) GO TO 14
      V(J,2)=SORVAL(K,L)
      GO TO 15
14    C(J,2)=SORVAL(K,L)
15    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE PERIOD (K,J)
      INTEGER TYPE
      COMMON /BLOCK3/ TYPE(200),VALUE(200),F(49,151),NOEL(7)
      COMMON /BLOCK7/ IDIMEN,ALPHA,FUNCT,NIC,NITER,H,EPS,NN,NB
      COMMON /BLOCK8/ V(200,2),C(200,2)
      COMMON /BLOCK10/ COND(200),IDEL,TSTART,TEND,NSTEP
      COMMON /BLOCK11/ SORVAL(5,100),TIMEPT(5,100),SNSOID(5,3),NNI(5)
      COMMON /BK13/ ISTEP,OLDVAL(5),SECVAL(5),OLDTIME(5),SECTIME(5),NNJ(
     15)
      COMMON /BLOCK14/ NP(5)
      NNN=NNI(K)
      TIME=TSTART+NIC*H
    1 T=TIME-NNJ(K)*(TIMEPT(K,NNN)-TSTART)
      IF (T.LE.SECTIME(K)) GO TO 2
      NP(K)=NP(K)+1
      IF (NP(K).GT.NNI(K)) GO TO 4
      NPR=NP(K)
      OLDTIME(K)=SECTIME(K)
      SECTIME(K)=TIMEPT(K,NPR)
      OLDVAL(K)=SECVAL(K)
      SECVAL(K)=SORVAL(K,NPR)
    2 IF (J.GE.NOEL(1)+1) GO TO 3
      V(J,2)=OLDVAL(K)+(SECVAL(K)-OLDVAL(K))*(T-OLDTIME(K))/(SECTIME(K)-
     1OLDTIME(K))
      GO TO 5
    3 C(J,2)=OLDVAL(K)+(SECVAL(K)-OLDVAL(K))*(T-OLDTIME(K))/(SECTIME(K)-
     1OLDTIME(K))
      GO TO 5
    4 NNJ(K)=NNJ(K)+1
      SECTIME(K)=TIMEPT(K,2)
      OLDTIME(K)=TIMEPT(K,1)
      SECVAL(K)=SORVAL(K,2)
      OLDVAL(K)=SORVAL(K,1)
      NP(K)=2
      GO TO 1
    5 RETURN
      END
```

```
      SUBROUTINE MATSET (A,NB,NN,NIC)
      INTEGER CNTSOR
      INTEGER TEMP
      INTEGER SORTYPE,CONTYPE
      INTEGER TYPE
      COMMON /BLOCK3/ TYPE(200),VALUE(200),F(49,151),NOEL(7)
      COMMON /BLOCK6/ TEMP(7),E(200),GRAD(200),CONST(200)
      COMMON /BLOCK8/ V(200,2),C(200,2)
      COMMON /BLOCK9/ SORTYPE(200),CONTYPE(200),KONBRAN(200)
      COMMON /BLOCK10/ COND(200),IDEL,TSTART,TEND,NSTEP
      COMMON /BLOCK19/ CNTSOR(200)
      COMMON /BLOCK22/ CK(200)
      COMMON /BLOCK25/ X(200),Y(200),Z(200,2)
      DIMENSION A(NB,NB)
      K=NN-1
      DO 4 I=1,K
      IF (TYPE(I).EQ.1HE) GO TO 1
      IF (TYPE(I).EQ.1HC) GO TO 2
      IF (TYPE(I).EQ.1HL) GO TO 3
      CK(I)=0.0
      GO TO 4
1     CK(I)=V(I,2)
      GO TO 4
2     IF (NIC.EQ.0) CK(I)=COND(I)
      IF (NIC.NE.0) CK(I)=V(I,1)+CONST(I)*C(I,1)
      GO TO 4
3     IF (NIC.EQ.0) CK(I)=0.0
      IF (NIC.NE.0) CK(I)=-V(I,1)-CONST(I)*C(I,1)
4     CONTINUE
      DO 8 I=NN,NB
      IF (TYPE(I).EQ.1HJ) GO TO 5
      IF (TYPE(I).EQ.1HC) GO TO 6
      IF (TYPE(I).EQ.1HL) GO TO 7
      CK(I)=0.0
      GO TO 8
5     CK(I)=C(I,2)
      GO TO 8
6     IF (NIC.EQ.0) CK(I)=0.0
      IF (NIC.NE.0) CK(I)=-C(I,1)-V(I,1)*CONST(I)
      GO TO 8
7     IF (NIC.EQ.0) CK(I)=COND(I)
      IF (NIC.NE.0) CK(I)=C(I,1)+V(I,1)*CONST(I)
8     CONTINUE
      DO 29 I=1,K
      IF (TYPE(I).EQ.1HE) GO TO 9
      IF (TYPE(I).EQ.1HV) GO TO 11
      IF (TYPE(I).EQ.1HC) GO TO 18
      IF (TYPE(I).EQ.1HR) GO TO 19
      IF (TYPE(I).EQ.1HL) GO TO 22
      IF (TYPE(I).EQ.1HI) GO TO 23
      GO TO 29
9     DO 10 J=1,NB
      A(I,J)=0.0
      IF (I.EQ.J) A(I,J)=1.0
10    CONTINUE
```

```
      GO TO 29
11    IF (CONTYPE(I).EQ.1HV) GO TO 12
      GO TO 15
12    M=CNTSOR(I)
      DO 13 J=1,K
      A(I,J)=-F(J,M-K)*CONST(I)
      IF (I.EQ.J) A(I,J)=1.0+A(I,J)
13    CONTINUE
      DO 14 J=NN,NB
      A(I,J)=0.0
14    CONTINUE
      GO TO 29
15    M=CNTSOR(I)
      DO 16 J=1,K
      A(I,J)=0.0
      IF (I.EQ.J) A(I,J)=1.0
16    CONTINUE
      DO 17 J=NN,NB
      A(I,J)=CONST(I)*F(M,J-K)
17    CONTINUE
      GO TO 29
18    IF (NIC.EQ.0) GO TO 9
19    DO 20 J=1,K
      A(I,J)=0.0
      IF (I.EQ.J) A(I,J)=1.0
20    CONTINUE
      DO 21 J=NN,NB
      A(I,J)=CONST(I)*F(I,J-K)
21    CONTINUE
      GO TO 29
22    IF (NIC.EQ.0) GO TO 9
      GO TO 19
23    IF (CONTYPE(I).EQ.1HI) GO TO 26
      M=CNTSOR(I)
      DO 24 J=1,K
      A(I,J)=-CONST(I)*F(J,M-K)
24    CONTINUE
      DO 25 J=NN,NB
      A(I,J)=-F(I,J-K)
25    CONTINUE
      GO TO 29
26    M=CNTSOR(I)
      DO 27 J=1,K
      A(I,J)=0.0
27    CONTINUE
      DO 28 J=NN,NB
      A(I,J)=-F(I,J-K)+CONST(I)*F(M,J-K)
28    CONTINUE
29    CONTINUE
      DO 49 I=NN,NB
      IF (TYPE(I).EQ.1HJ) GO TO 30
      IF (TYPE(I).EQ.1HI) GO TO 32
      IF (TYPE(I).EQ.1HC) GO TO 38
      IF (TYPE(I).EQ.1HR) GO TO 39
      IF (TYPE(I).EQ.1HL) GO TO 42
```

```
      IF (TYPE(I).EQ.1HV) GO TO 43
      GO TO 49
30    DO 31 J=1,NB
      A(I,J)=0.0
      IF (I.EQ.J) A(I,J)=1.0
31    CONTINUE
      GO TO 49
32    IF (CONTYPE(I).EQ.1HI) GO TO 35
      M=CNTSOR(I)
      DO 33 J=1,K
      A(I,J)=-F(J,M-K)*CONST(I)
33    CONTINUE
      DO 34 J=NN,NB
      A(I,J)=0.0
      IF (I.EQ.J) A(I,J)=1.0
34    CONTINUE
      GO TO 49
35    DO 36 J=1,K
      A(I,J)=0.0
36    CONTINUE
      M=CNTSOR(I)
      DO 37 J=NN,NB
      A(I,J)=F(M,J-K)*CONST(I)
      IF (I.EQ.J) A(I,J)=A(I,J)+1.0
37    CONTINUE
      GO TO 49
38    IF (NIC.EQ.0) GO TO 30
39    DO 40 J=1,K
      A(I,J)=-F(J,I-K)*CONST(I)
40    CONTINUE
      DO 41 J=NN,NB
      A(I,J)=0.0
      IF (I.EQ.J) A(I,J)=1.0
41    CONTINUE
      GO TO 49
42    IF (NIC.EQ.0) GO TO 30
      GO TO 39
43    IF (CONTYPE(I).EQ.1HI) GO TO 46
      M=CNTSOR(I)
      DO 44 J=1,K
      A(I,J)=F(J,I-K)-F(J,M-K)*CONST(I)
44    CONTINUE
      DO 45 J=NN,NB
      A(I,J)=0.0
45    CONTINUE
      GO TO 49
46    DO 47 J=1,K
      A(I,J)=F(J,I-K)
47    CONTINUE
      M=CNTSOR(I)
      DO 48 J=NN,NB
      A(I,J)=CONST(I)*F(M-K,J-K)
48    CONTINUE
49    CONTINUE
      CALL MATINV (A,NB,DETERM,X,Y,Z)
```

```
      DO 50 I=1,K
      V(I,2)=0.0
      DO 50 J=1,NB
50    V(I,2)=V(I,2)+A(I,J)*CK(J)
      DO 51 I=NN,NB
      C(I,2)=0.0
      DO 51 J=1,NB
51    C(I,2)=C(I,2)+A(I,J)*CK(J)
      RETURN
      END
```

```
      SUBROUTINE MATINV (A,N,DETERM,PIVOT,IPIVOT,INDEX)
C
C     MATRIX INVERSION WITH ACCOMPANYING SOLUTION OF LINEAR EQUATIONS
C
      DIMENSION IPIVOT(N), PIVOT(N), INDEX(N,2), A(N,N)
      EQUIVALENCE (IROW,JROW), (ICOLUM,JCOLUM), (AMAX,T,SWAP)
C
C     INITIALIZATION
C
1     DETERM=1.0
      DO 2 J=1,N
2     IPIVOT(J)=0
      DO 15 I=1,N
C
C     SEARCH FOR PIVOT ELEMENT
C
      AMAX=0.0
      DO 7 J=1,N
      IF (IPIVOT(J)-1) 3,7,3
3     DO 6 K=1,N
      IF (IPIVOT(K)-1) 4,6,19
4     IF (ABS(AMAX)-ABS(A(J,K))) 5,6,6
5     IROW=J
      ICOLUM=K
      AMAX=A(J,K)
6     CONTINUE
7     CONTINUE
      IF (AMAX) 8,20,8
8     IPIVOT(ICOLUM)=IPIVOT(ICOLUM)+1
C
C     INTERCHANGE ROWS TO PUT PIVOT ELEMENT ON DIAGONAL
C
      IF (IROW-ICOLUM) 9,11,9
9     DETERM=-DETERM
      DO 10 L=1,N
      SWAP=A(IROW,L)
      A(IROW,L)=A(ICOLUM,L)
10    A(ICOLUM,L)=SWAP
11    INDEX(I,1)=IROW
      INDEX(I,2)=ICOLUM
      PIVOT(I)=A(ICOLUM,ICOLUM)
      DETERM=DETERM*PIVOT(I)
C
C     DIVIDE PIVOT ROW BY PIVOT ELEMENT
C
      A(ICOLUM,ICOLUM)=1.0
      DO 12 L=1,N
12    A(ICOLUM,L)=A(ICOLUM,L)/PIVOT(I)
C
C     REDUCE NON-PIVOT ROWS
C
      DO 15 L1=1,N
      IF (L1-ICOLUM) 13,15,13
13    T=A(L1,ICOLUM)
      A(L1,ICOLUM)=0.0
```

```
      DO 14 L=1,N
   14 A(L1,L)=A(L1,L)-A(ICOLUM,L)*T
   15 CONTINUE
C
C     INTERCHANGE COLUMNS
C
      DO 18 I=1,N
      L=N+1-I
      IF (INDEX(L,1)-INDEX(L,2)) 16,18,16
   16 JROW=INDEX(L,1)
      JCOLUM=INDEX(L,2)
      DO 17 K=1,N
      SWAP=A(K,JROW)
      A(K,JROW)=A(K,JCOLUM)
      A(K,JCOLUM)=SWAP
   17 CONTINUE
   18 CONTINUE
   19 RETURN
   20 DETERM=0.
      RETURN
      END
```

```
      SUBROUTINE READOUT
      INTEGER TEMP
      INTEGER SORTYPE,CONTYPE
      INTEGER TYPE
      COMMON /BLOCK3/ TYPE(200),VALUE(200),F(49,151),NOEL(7)
      COMMON /BLOCK5/ IOUT(200),NOUT(200),ITEST(200)
      COMMON /BLOCK6/ TEMP(7),E(200),GRAD(200),CONST(200)
      COMMON /BLOCK7/ IDIMEN,ALPHA,FUNCT,NIC,NITER,H,EPS,NN,NB
      COMMON /BLOCK8/ V(200,2),C(200,2)
      COMMON /BLOCK9/ SORTYPE(200),CONTYPE(200),KONBRAN(200)
      COMMON /BLOCK10/ COND(200),IDEL,TSTART,TEND,NSTEP
      COMMON /BLOCK15/ NIT,JOUT,NGRAPH,NALLOUT
      COMMON /BLOCK16/ GRAF(200,5),JGRAPH(5),IGRAPH(5),NPRINT,SCALE,NITT
      COMMON /BLOCK18/ VAROUT(200)
      N=NN-1
      DO 4 I=1,JOUT
      K=NOUT(I)
      IF (NOUT(I).LT.NN) GO TO 2
      IF (ITEST(I).EQ.1) GO TO 1
      VAROUT(I)=V(K,2)
      GO TO 4
1     VAROUT(I)=C(K,2)/SCALE
      GO TO 4
2     IF (ITEST(I).EQ.0) GO TO 3
      VAROUT(I)=C(K,2)/SCALE
      GO TO 4
3     VAROUT(I)=V(K,2)
4     CONTINUE
      TIME=H*NIC+TSTART
      IF (NIC.EQ.0) PRINT 6
      PRINT 7, TIME
      DO 5 I=1,JOUT
      A=7HVOLTAGE
      IF (ITEST(I).EQ.1) A=7HCURRENT
      PRINT 8, IOUT(I),A,VAROUT(I)
5     CONTINUE
      RETURN
C
6     FORMAT (1H1)
7     FORMAT (/////,10X,*TIME = *,E15.5)
8     FORMAT (1H0,10X,*BRANCH*,I5,5X,A7,10X,E15.5)
      END
```

```
      SUBROUTINE GRAPH
      INTEGER TYPE
      COMMON /BLOCK1/ ANP(200),NP(200)
      COMMON /BLOCK3/ TYPE(200),VALUE(200),F(49,151),NOEL(7)
      COMMON /BLOCK4/ ITBRAN(200),LEAVT(200),LENTT(200)
      COMMON /BLOCK7/ IDIMEN,ALPHA,FUNCT,NIC,NITER,H,EPS,NN,NB
      COMMON /BLOCK8/ V(200,2),C(200,2)
      COMMON /BLOCK10/ COND(200),IDEL,TSTART,TEND,NSTEP
      COMMON /BLOCK15/ NIT,JOUT,NGRAPH,NALLOUT
      COMMON /BLOCK16/ GRAF(200,5),JGRAPH(5),IGRAPH(5),NPRINT,SCALE,NITT
      COMMON /BLOCK21/ NST(5)
1     DO 3 I=1,NGRAPH
      K=NST(I)
      IF (JGRAPH(I).EQ.0) GO TO 2
      GRAF(NPRINT,I)=C(K,2)/SCALE
      GO TO 3
2     GRAF(NPRINT,I)=V(K,2)
3     CONTINUE
      IF (NPRINT.LT.200.AND.(NITER-NIC).GE.NSTEP) RETURN
      DO 9 I=1,NGRAPH
      PRINT 11
      A=7HVOLTAGE
      IF (JGRAPH(I).EQ.1) A=7HCURRENT
      PRINT 10, IGRAPH(I),A
      AMAX=GRAF(1,I)
      AMAX=ABS(AMAX)
      DO 4 J=2,NPRINT
      A=GRAF(J,I)
      A=ABS(A)
      IF (A.GT.AMAX) AMAX=A
4     CONTINUE
      DO 5 J=1,NPRINT
      NP(J)=GRAF(J,I)*50.0/AMAX+51.0
5     CONTINUE
      DO 6 J=1,101
      ANP(J)=1H.
6     CONTINUE
      PRINT 12, (ANP(J),J=1,101)
      DO 7 J=1,102
      ANP(J)=1H
7     CONTINUE
      ANP(51)=1H.
      DO 8 KK=1,NPRINT
      LL=NP(KK)
      IF (LL.GE.51) LL=LL+1
      ANP(LL)=1H+
      TIME=TSTART+NSTEP*KK*H-H
      PRINT 13, TIME,GRAF(KK,I),(ANP(J),J=1,102)
      ANP(LL)=1H
8     CONTINUE
9     CONTINUE
      NGRAPH=201
      RETURN
C
10    FORMAT (1H0,30X,*BRANCH*,I4,5X,A7)
```

```
11    FORMAT (1H1)
12    FORMAT  (1H0,2X,*TIME*,10X,*VALUE*,6X,101A1)
13    FORMAT (1HZ,E11.4,2X,E11.4,3X,102A1)
      END
```

```
      SUBROUTINE ALLOUT
      COMMON /BLOCK7/ IDIMEN,ALPHA,FUNCT,NIC,NITER,H,EPS,NN,NB
      COMMON /BLOCK8/ V(200,2),C(200,2)
      COMMON /BLOCK10/ COND(200),IDEL,TSTART,TEND,NSTEP
      COMMON /BLOCK16/ GRAF(200,5),JGRAPH(5),IGRAPH(5),NPRINT,SCALE,NITT
      COMMON /BLOCK18/ A(200)
      DO 1 I=NN,NB
      A(I)=C(I,2)/SCALE
1     CONTINUE
      N=NN-1
      TIME=(TSTART+H*NIC)
      PRINT 2, TIME
      PRINT 3, (V(I,2),I=1,N)
      PRINT 4, (A(I),I=NN,NB)
      RETURN
C
2     FORMAT(1H0,///,*   TIME = *,E15.5)
3     FORMAT(1H0,*THESE ARE THE TREE BRANCH VOLTAGES*,//,10(3X,E12.5))
4     FORMAT (1H0, * THESE ARE THE LINK CURRENTS *,//,10(3X,E12.5))
      END
```

APPENDIX E

General Comments Regarding Use of CANINE:

1. Number of nodes cannot exceed 50.

2. Number of branches cannot exceed 200.

3. Cannot exceed 5 arbitrary or periodic independent sources, and cannot exceed 5 sinusoidal or exponential sources. There is no limit on the number of constant independent sources. We can have at most 100 time points to describe an arbitrary independent source.

4. Cannot exceed 5 graphical outputs.

5. There is an internal limit of 200 output points per graph.

6. Branch and node numbering is arbitrary, but branch numbers cannot exceed three digits, and node numbers cannot exceed two digits.

7. Core requirements:

Since our inversion scheme requires the storing of an NB $\times$ NB matrix, it would, in general, be inconvenient to always dimension the above matrix 200 $\times$ 200.

Thus, to save core, one need only replace the DIMENSION statement in the MAIN program by

```
DIMENSION HH(N,N)
```

where N is the maximum number of branches to be encountered in the network (s) to be solved.

With the above dimensioning procedure, the core requirements for CANINE are, approximately,

$$27000 + N^2 \text{ (decimal)}$$

storage locations.

8. There must be at least one branch which is not an independent source, in any given network.

9. CANINE can handle zero valved R's, L's and C's as long as there are no loops of zero valved inductances or resistances (short circuits), and no cutsets of zero valved capacitances (open circuits). In this context, independent voltage sources are equivalent to short circuits, and independent current sources are equivalent to open circuits.

10. H, the time increment, cannot be zero for networks containing reactive elements (L's and C's).

11. CANINE can be used for batch processing; simply stack data card sets, for the networks to be solved, one behind the other, ensuring that the last card in the stack is a blank card.

12. Zero is not a valid branch or node number.

13. The output voltage and current units are always volts and amperes respectively.

BIBLIOGRAPHY

1. Paul M. Russo: "CANDO, Computer Analysis of Networks with Design Orientation," University of California Memorandum No. ERL-M251.