ERRATA

Corrections to listing of CANDO: ERL-M251

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

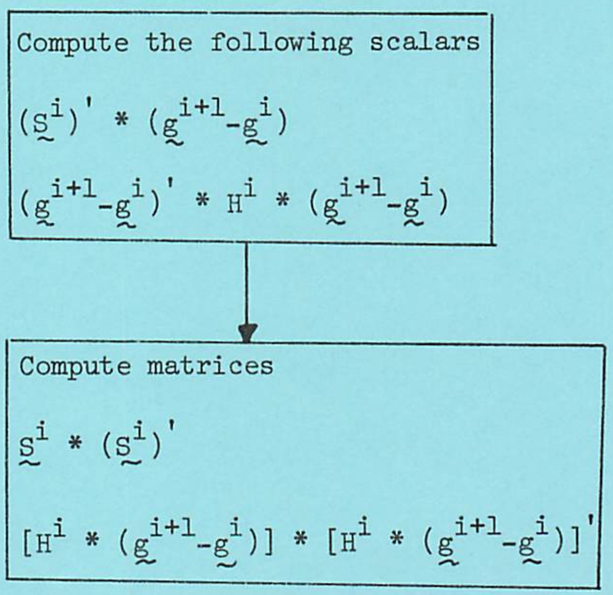
```
      62   I = I - 1
```

ERL-M251 <u>ERRATUM:</u>

Page 37, the expression for $\underset{\wedge}{H}^{i+1}$ should read:

$$\underset{\wedge}{H}^{i+1} = \underset{\wedge}{H}^{i} - \frac{\underset{\wedge}{s}^{i} * (\underset{\wedge}{s}^{i})'}{(\underset{\sim}{s}^{i})' * (\underset{\sim}{g}^{i+1}-\underset{\sim}{g}^{i})} - \frac{[\underset{\sim}{H}^{i} * (\underset{\sim}{g}^{i+1}-\underset{\sim}{g}^{i})] * [\underset{\sim}{H}^{i} * (\underset{\sim}{g}^{i+1}-\underset{\sim}{g}^{i})]'}{(\underset{\sim}{g}^{i+1}-\underset{\sim}{g}^{i})' * H^{i} * (\underset{\sim}{g}^{i+1}-\underset{\sim}{g}^{i})}$$

Page 38, the indicated blocks should contain the following



Compute the following scalars

$(\underset{\sim}{s}^{i})' * (\underset{\sim}{g}^{i+1}-\underset{\sim}{g}^{i})$

$(\underset{\sim}{g}^{i+1}-\underset{\sim}{g}^{i})' * H^{i} * (\underset{\sim}{g}^{i+1}-\underset{\sim}{g}^{i})$

Compute matrices

$\underset{\sim}{s}^{i} * (\underset{\sim}{s}^{i})'$

$[H^{i} * (\underset{\sim}{g}^{i+1}-\underset{\sim}{g}^{i})] * [H^{i} * (\underset{\sim}{g}^{i+1}-\underset{\sim}{g}^{i})]'$

CANDO (Computer Analysis of Networks With Design Orientation)

by

Paul M. Russo

Memorandum No. ERL-M251

24 July 1968

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

ABSTRACT:

This report describes the operation and use of CANDO (Computer Analysis of Networks with Design Orientation), a time domain, design oriented, analysis program for linear time-invariant networks. The networks may contain dependent and independent sources, 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 CANDO, nonindependent source tree voltages and nonindependent source link currents form a basis set of variables. This formulation yields the automatic satisfaction of Kirchhoff's laws and also allows us to solve an optimally low order system of equations. A numerical integration formula reduces the system to a set of simultaneous algebraic equations of the form

$$E_i = 0 \quad ,$$

which are solved by minimizing a performance criterion ϵ defined by

$$\epsilon \equiv \frac{1}{2} \sum_i E_i^2$$

CANDO utilizes the Fletcher-Powell and Rohrer search minimization algorithms in minimizing ϵ.

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 CANDO is written in FORTRAN IV for the CDC 6400 system.

ACKNOWLEDGMENTS:

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:

In computerized fully automated network design, where one performs several network analyses at each iterative step of the design process, one has a need for an analysis program which can be made arbitrarily accurate, and whose execution time is a function of the accuracy desired.

CANDO satisfies the above requirement, since one can specify either a maximum permissible value of the performance criterion ϵ, and/or a maximum number of minimization iterative steps, per time point.

The iterative scheme utilized by CANDO is such that it can be easily extended to networks containing nonlinear and time-varying elements, by the simple addition of corresponding subroutines.

Section I describes the iterative scheme utilized in solving both the initial condition problem and the general time point problem. Section II describes the MAIN program and subroutines associated with CANDO. The appendices describe the use of CANDO in solving network problems, and the way in which dependent sources must be modeled.

SECTION I: Network Analysis

Theory:

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

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

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

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

where $\underset{\sim}{Q}$ is the fundamental cutset matrix and $\underset{\sim}{B}$ is the fundamental loop matrix. Upon the selection of an appropriate 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 this numbering scheme, we may partition $\underset{\sim}{Q}$ and $\underset{\sim}{B}$ as follows:

$$\underset{\sim}{Q} = [\underset{\sim}{I} \mid \underset{\sim}{F}] \quad \text{and} \quad \underset{\sim}{B} = [-\underset{\sim}{F}' \mid \underset{\sim}{I}]$$

where $\underset{\sim}{I}$ is the identity matrix and hence we have, from Kirchhoff's laws,

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

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

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

Our tree-picking algorithms are discussed in Subroutine PT; 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 ERROR), our set of linear equations reduces to

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

and the dimension of the system is NB minus the number of independent sources.

This set of equations is solved by minimizing a performance criterion ϵ defined by

$$\epsilon = \frac{1}{2} \sum E_i^2 = \frac{1}{2} \underset{\sim}{E}' \cdot \underset{\sim}{E}$$

The minimization is performed by using the Fletcher Powell algorithm to give us a direction along which to minimize (see Subroutine FLPOW), and using Rohrer search to find the optimum step size along that direction (see Subroutine FNDALPH).

This iterative scheme is continued until either the error is sufficiently small or we have gone the desired number of steps. (See Program MAIN).

Solving the Initial Condition Problem:

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 proper tree is selected according to the algorithm described in subroutine PT, allowing us to reduce the order of the system. This algorithm maximizes the number of capacitances in tree branches and the number of inductances in links. In general, 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 upon the selection of the tree voltages and link currents to be a basis set of variables, with the following relations emerging:

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

and

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

where the subscripts ℓ and t refer to links and tree branches respectively, and where $\underset{\wedge}{Q} = [\underset{\wedge}{I} \mid \underset{\wedge}{F}]$ is the fundamental cutset matrix based on the <u>proper</u> tree.

To further reduce the dimension of the system, we do not treat independent sources as variables, but rather require independent voltage sources to be tree branches, and independent current sources to be links.

Furthermore, 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 equal to the initial condition. Tree inductance voltages and link capacitance currents are set equal to zero, and their initial 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.

Thus the system of equations we are solving is one corresponding to a network consisting only of independent sources, dependent sources and resistances.

The solution is obtained in a way analogous to the solution of the equations associated with the "general time point problem" (see next section), and will not be detailed here.

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 $\underset{\sim}{i}_2$ and $\underset{\sim}{v}_2$, 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 (see subroutine PT), allowing us to reduce the order of the system, and hence to improve the convergence of our iterative *minimization* 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 upon the selection of the tree branch voltages and link currents to be a basis set of variables.

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

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

where $\underset{\sim}{Q} = [\underset{\sim}{I} \mid \underset{\sim}{F}]$ is the fundamental cutset matrix associated with our *optimal* tree.

Similarly,

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

where $\underset{\sim}{F}'$ is the transpose of $\underset{\sim}{F}$. We thus wish to solve the set of equations

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

which are our branch errors (see subroutine ERROR).

To further reduce the dimension of the space, we do not treat independent source values as variables. Notice that this requirement dictates that independent voltage sources be tree branches, and that independent current sources be links.

The solution to $\underset{\sim}{E}(\underset{\sim}{i}_\ell, \underset{\sim}{v}_t) = \underset{\sim}{0}$ is obtained by minimizing a performance criterion ϵ, given by

$$\epsilon = \sum_i E_i^{\,2}$$

Note that we have an exact solution when $\varepsilon = 0$.

ε is minimized via the Fletcher-Powell minimization algorithm(1) (see subroutine FLPOW) with Rohrer search (see subroutine FNDALPH) being utilized to find the optimal step size along the direction generated by the Fletcher-Powell algorithm.

Once $\underset{\sim}{v}_t$ and $\underset{\sim}{i}_\ell$ are known, we have

$$\underset{\sim}{i}_2 = \left[\begin{array}{c} -\underset{\sim}{F} \\ \hline \underset{\sim}{I} \end{array}\right] \underset{\sim}{i}_\ell$$

and

$$\underset{\sim}{v}_2 = \left[\begin{array}{c} \underset{\sim}{I} \\ \hline \underset{\sim}{F}' \end{array}\right] \underset{\sim}{v}_t \quad ,$$

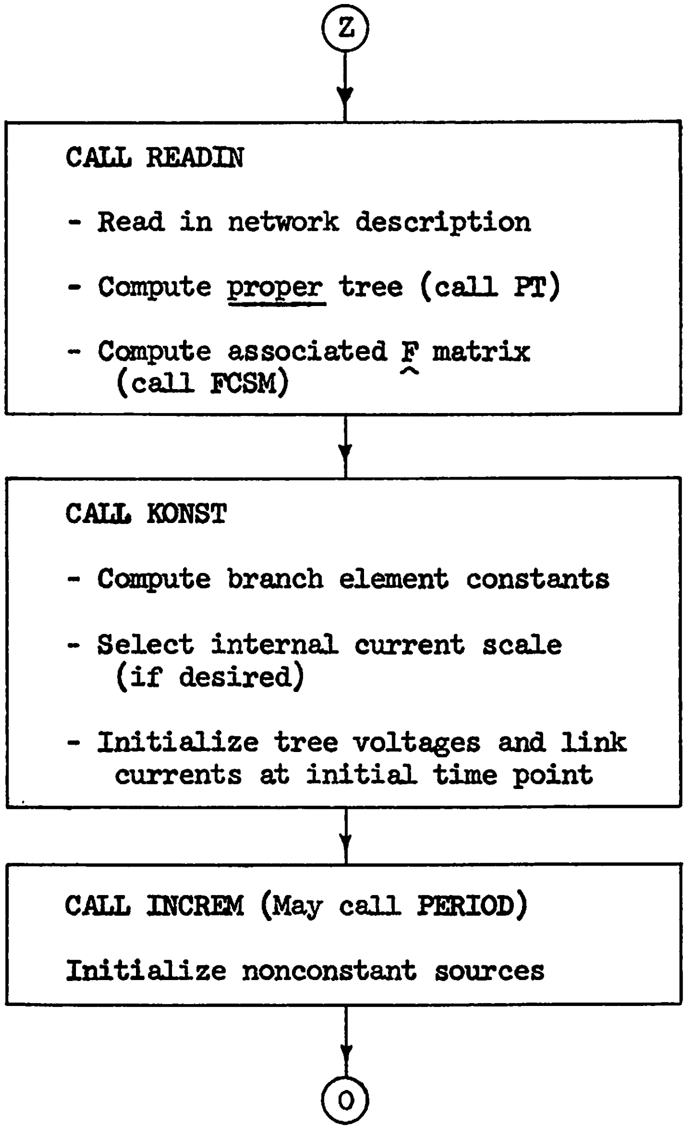
and hence, we proceed to the subsequent time point in exactly the same way.

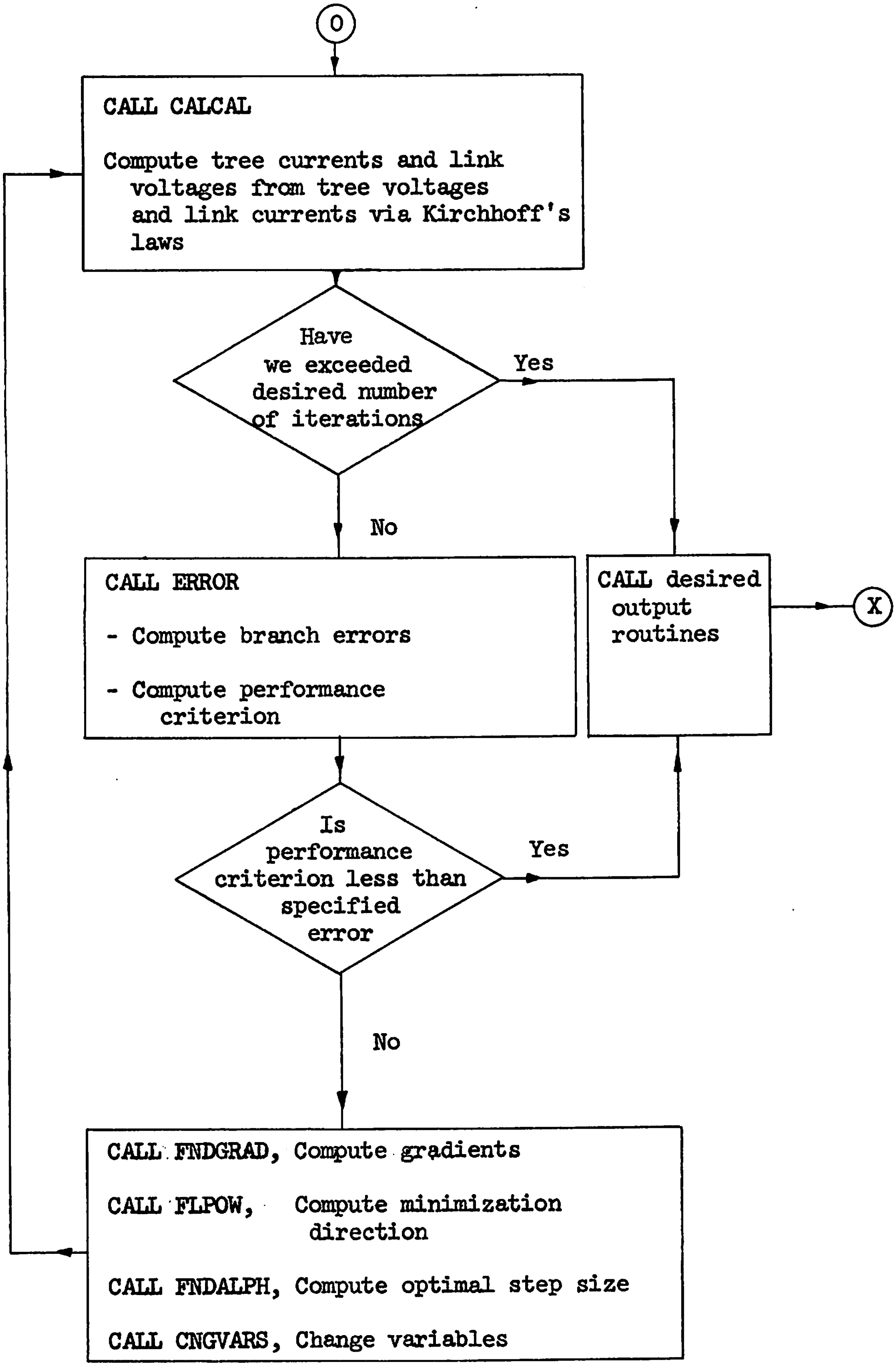
SECTION II: PROGRAM CANDO

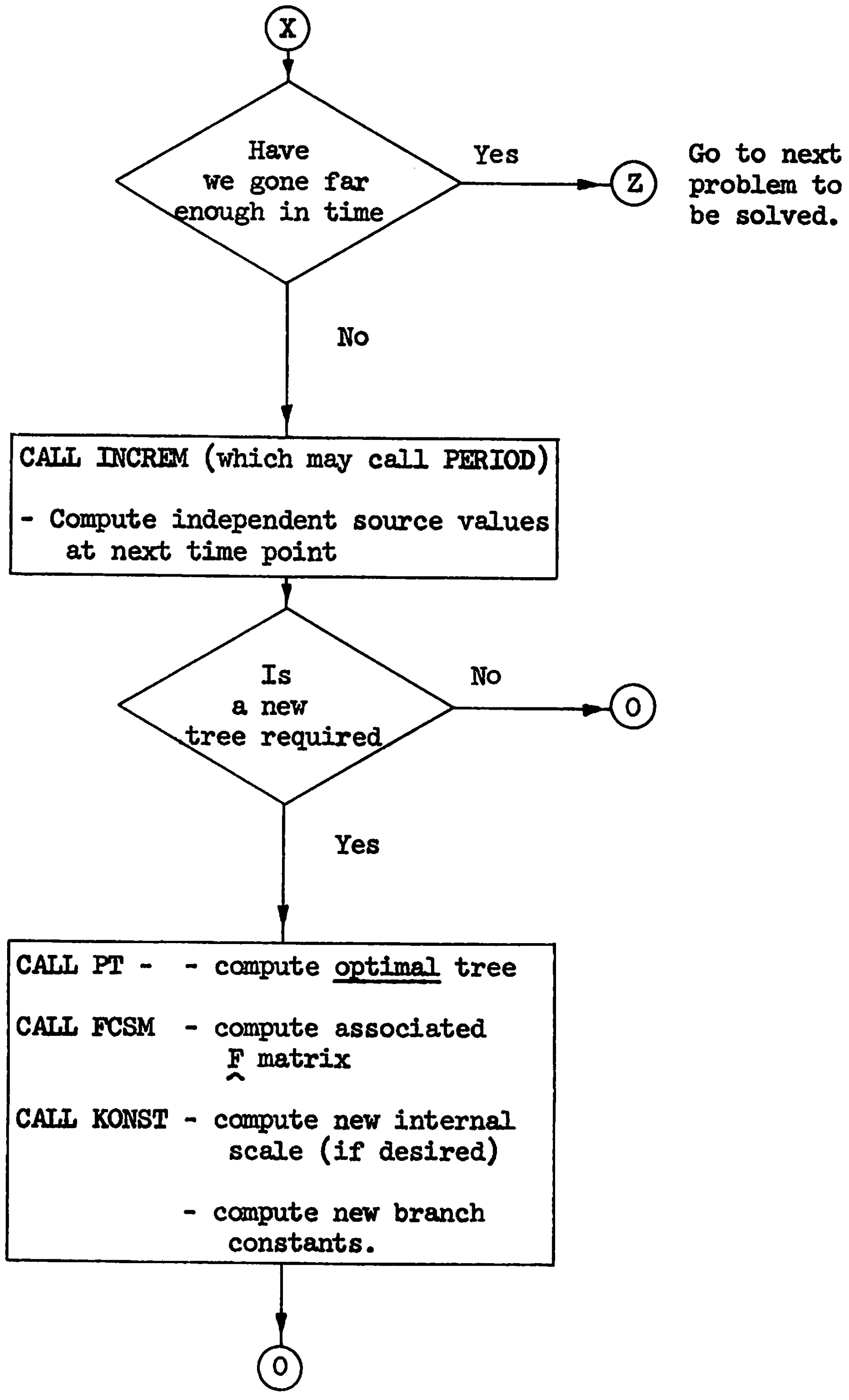
MAIN program:

The MAIN section of program CANDO ensures that the desired subroutines are called in the correct sequence. It solves both the initial condition problem and (when required) the general time point problem, in the way discussed in Section I under Theory. It also controls the calls to the output subroutines.

BLOCK DIAGRAM: - MAIN PROGRAM


Z
CALL READIN
- Read in network description
- Compute proper tree (call PT)
- Compute associated $\underset{\wedge}{F}$ matrix (call FCSM)
CALL KONST
- Compute branch element constants
- Select internal current scale (if desired)
- Initialize tree voltages and link currents at initial time point
CALL INCREM (May call PERIOD)
Initialize nonconstant sources
O


O
CALL CALCAL
Compute tree currents and link voltages from tree voltages and link currents via Kirchhoff's laws
Have we exceeded desired number of iterations
Yes
No
CALL ERROR
- Compute branch errors
- Compute performance criterion
CALL desired output routines
X
Is performance criterion less than specified error
Yes
No
CALL FNDGRAD, Compute gradients
CALL FLPOW, Compute minimization direction
CALL FNDALPH, Compute optimal step size
CALL CNGVARS, Change variables


X
Have we gone far enough in time
Yes
Z
No
CALL INCREM (which may call PERIOD)
- Compute independent source values at next time point
Is a new tree required
No
O
Yes
CALL PT - - compute optimal tree
CALL FCSM - compute associated $\hat{F}$ matrix
CALL KONST - compute new internal scale (if desired)
- compute new branch constants.
O

Go to next problem to be solved.

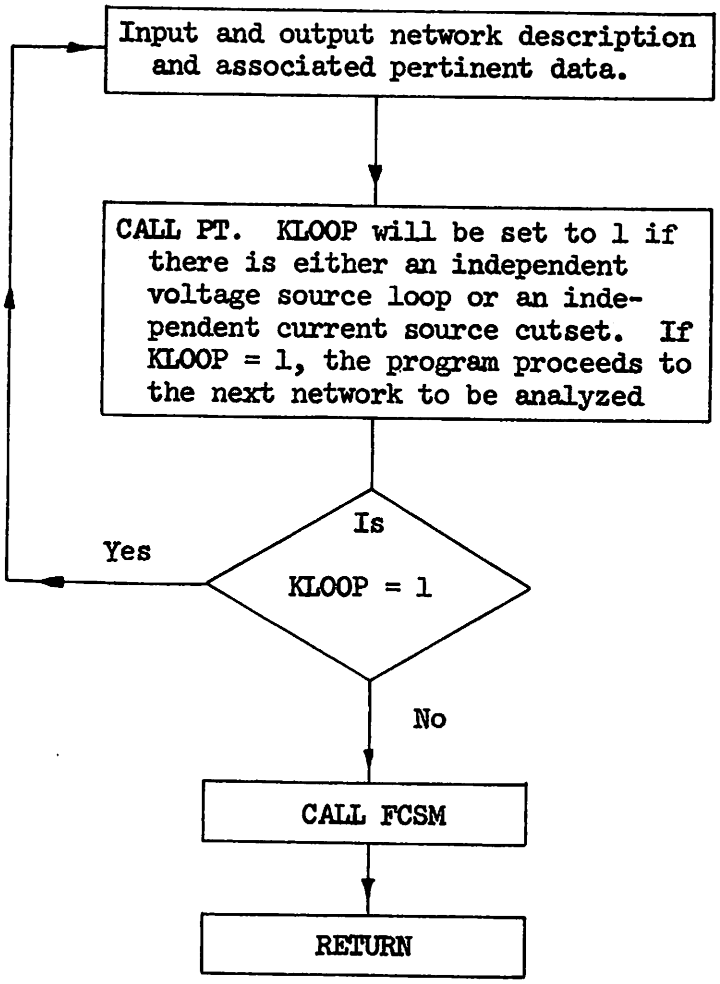
SUBROUTINE READIN:

This subroutine reads in the complete network description along with all pertinent output and control specifications. As a data card check, the given network description and the pertinent control specifications are printed out.

For a detailed description of the variables read in, see the section entitled "Data Cards for CANDO."

This subroutine also calls subroutine PT which computes an appropriate tree to solve the initial condition problem. If no independent voltage source loops or independent current source cutsets exist, subroutine FCSM is called, otherwise the program proceeds to the next set of data. Subroutine FCSM computes the nontrivial portion of the fundamental cutset matrix.

BLOCK DIAGRAM:


Input and output network description and associated pertinent data.
CALL PT. KLOOP will be set to 1 if there is either an independent voltage source loop or an independent current source cutset. If KLOOP = 1, the program proceeds to the next network to be analyzed
Is KLOOP = 1
Yes
No
CALL FCSM
RETURN

SUBROUTINE READIN

SUBROUTINE PT

The first time it is called, this subroutine computes a proper tree for the initial condition problem; on subsequent calls in the same problem, it computes an optimal tree. The distinguishing features of the above two tree-picking schemes will be enunciated below.

If the given network has NN nodes and NB branches, it will have NN-1 tree branches and NB - NN + 1 cotree links.

A proper tree consists of the first NN-1 branches which do not form loops, with the following selection priority scheme (in descending order)

1. independent voltage sources
2. dependent voltage sources
3. capacitances
4. resistances in ascending order of value
5. inductances
6. dependent current sources
7. independent current sources

An independent voltage source appearing in a link indicates the presence of a loop of independent voltage sources. An independent current source appearing in a tree branch indicates an independent current source cutset. In both of these cases, an error diagnostic will be printed out, and the program will proceed to the next problem.

An optimal tree consists of the first NN-1 branches which do not form loops, with the following selection priority scheme (in descending order)

1. independent voltage sources
2. equivalent resistances in ascending order of value
3. independent current sources

The equivalent resistance values are given by the following prescription:

Controlled voltage source → 10^{-50}

Resistance (R) → R

Capacitance (C) → $\frac{2H}{C}$

Inductance (L) → $\frac{L}{2H}$

Controlled current source → 10^{50},

where H is the specified integration step size for use with the trapezoidal rule. If C = 0, the corresponding equivalent resistance is set to 10^{51}.

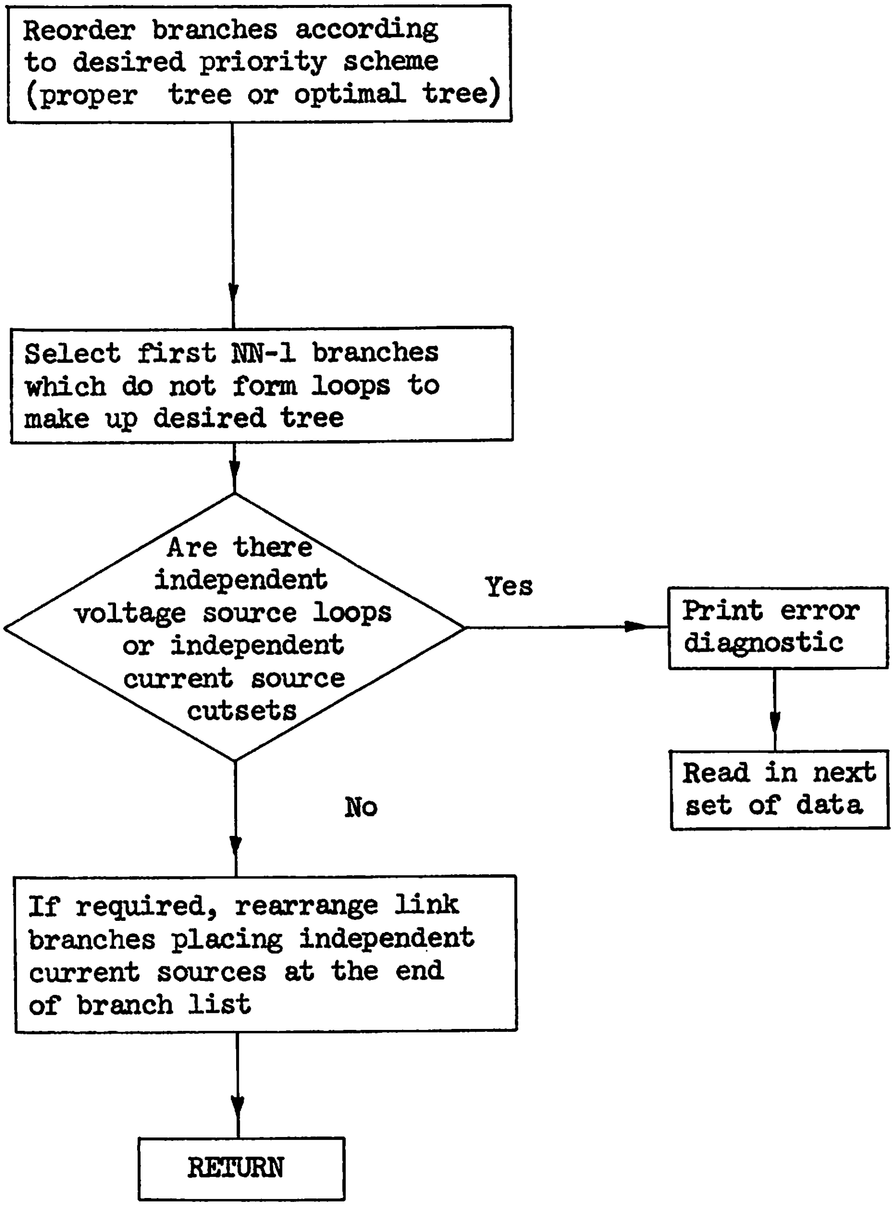
The internal branch numbering scheme is such that the first NN-1 branches are tree branches, the remaining being cotree links. This modification allows us to subsequently store only a portion of the fundamental cutset matrix.

Note that in the selection of a proper tree, the independent current sources are placed at the end of the branch list. In the subsequent selection of an optimal tree, the independent current sources are not moved around and hence need not be rearranged.

This subroutine, when desired, prints out the selected tree and the rearranged network.

Note that when a network contains only sources and resistances (i.e., no reactive elements), the proper and optimal trees coincide.

BLOCK DIAGRAM


Reorder branches according to desired priority scheme (proper tree or optimal tree)
Select first NN-1 branches which do not form loops to make up desired tree
Are there independent voltage source loops or independent current source cutsets
Yes
Print error diagnostic
Read in next set of data
No
If required, rearrange link branches placing independent current sources at the end of branch list
RETURN

SUBROUTINE PT

SUBROUTINE FCSM (Fundamental Cutset Matrix):

Given a connected network, and a specified tree, there exists a unique (NN-1 × NB) fundamental cutset matrix $\underset{\wedge}{Q}$, where NN and NB are the number of nodes and the number of branches in the network, respectively.

Kirchhoff's current law is then given by:

$$\underset{\wedge}{Q}\underset{\wedge}{i}_b = \underset{\wedge}{0} \quad ,$$

where $\underset{\wedge}{i}_b$ is the branch current vector.

If the branch numbering is selected such that the first NN-1 branches are tree branches (see subroutine PT), then, $\underset{\wedge}{Q}$ can be partitioned as follows:

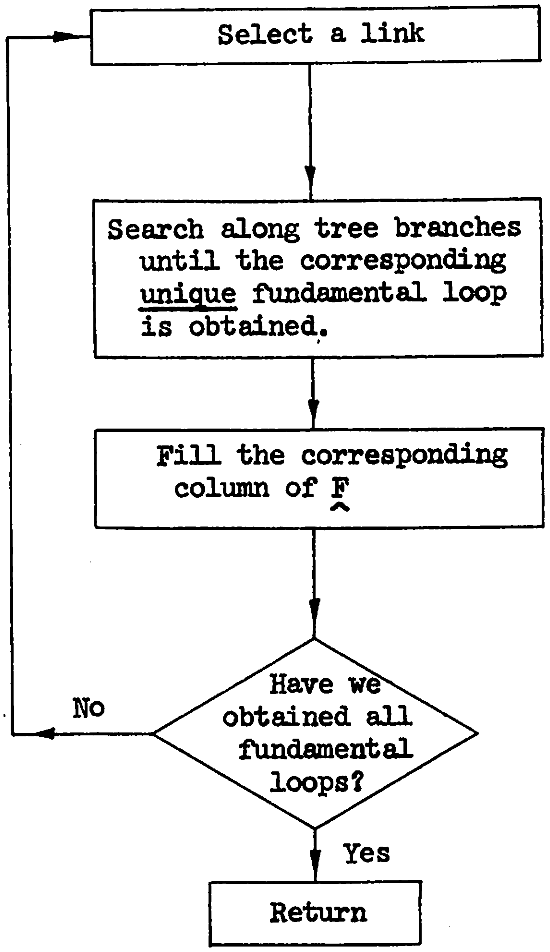
$$\underset{\wedge}{Q} = [\underset{\wedge}{I}|\underset{\wedge}{F}]$$

and only the (NN-1 × NB-NN + 1) $\underset{\wedge}{F}$ matrix need be stored, resulting in a substantial saving of computer memory.

With each link, of our connected network, is associated a <u>unique</u> fundamental loop, consisting of the link itself and sufficient tree branches required to close the loop. Each column of $\underset{\wedge}{F}$ corresponds to such a fundamental loop. This intermediate scheme is used to compute $\underset{\wedge}{F}$.

If desired, this subroutine can output the $\underset{\wedge}{F}$ portion of the fundamental cutset matrix.

BLOCK DIAGRAM:


Select a link
Search along tree branches until the corresponding unique fundamental loop is obtained.
Fill the corresponding column of $\hat{F}$
Have we obtained all fundamental loops?
No
Yes
Return

SUBROUTINE FCSM

SUBROUTINE KONST:

This subroutine computes the branch element constants required in the branch relations. It initializes tree voltages and link currents, and, when desired (NSCALE $\neq$ 1), it selects a new current scale factor each time a new tree is picked.

There are no constants associated with independent sources.

The constants are computed according to the prescription given below, where H denotes the integration time step, in seconds, S denotes the scale factor which selects the desired current unit (e.g., S = 1000 implies current unit is milliamps), and λ_i denotes the branch element constant associated with the ith branch. See subroutine ERROR for the form of the branch relations.

Dependent voltage sources:

a) voltage controlled case

$$\lambda_i = K_i$$

where K_i is the ith coupling constant and is unitless.

b) current controlled case

$$\lambda_i = \frac{K_i}{S}$$

where K_i is the ith coupling constant, in ohms.

Capacitances:

a) tree branch case

$$\lambda_i = \frac{H}{2 * C_i * S}$$

b) link case

$$\lambda_i = \frac{2 * C_i * S}{H}$$

where C_i is the ith capacitance value, in farads.

Resistances:

a) tree branch case

$$\lambda_i = \frac{R_i}{S}$$

b) link case

$$\lambda_i = \frac{S}{R_i}$$

where R_i is the ith resistance value in ohms.

Inductances:

a) tree branch case

$$\lambda_i = \frac{2 * L_i}{H * S}$$

b) link case

$$\lambda_i = \frac{H * S}{2 * L_i}$$

where L_i is the ith inductance value in henrys.

Dependent current sources:

a) voltage controlled case

$$\lambda_i = K_i * S$$

where K_i is the ith coupling constant, in ohms.

b) current controlled case

$$\lambda_i = K_i$$

where K_i is the ith coupling constant and is unitless.

Initialization:

The first time that subroutine KONST is called it initializes tree voltages and link currents as follows:

tree voltages: Tree inductance and tree dependent current source voltages are set equal to zero, the remaining being set to the given initial conditions.

link currents: link capacitance and link dependent voltage source currents are set equal to zero, the remaining being set to the given initial conditions.

It should be noted that the above initialization scheme makes sense in light of the fact that the associated tree is a proper tree.

Automatic computation of scale factor:

When desired, this subroutine will compute a new current scale factor each time a new tree is picked.

The algorithm, for selecting the optimum scale factor, for both proper and optimal trees, is as follows:

1. select smallest link resistance value (say R_i)
2. set internal current unit to be $\frac{\text{amps}}{R_i}$ where R_i is in ohms, e.g., if R_i = 14800 ohms, the internal current unit will be $\frac{1}{14.8}$ milliamps.

If our tree selection yields the fact that there are no link resistances, the scale factor is set equal to the maximum tree resistance. If our network contains no resistances, the scale factor is automatically set equal to 1.0.

SUBROUTINE CALCAL:

This subroutine computes the tree branch currents $\underset{\sim}{i}_t$ and the link voltages$\underset{\sim}{v}_\ell$ from the link currents $\underset{\sim}{i}_\ell$ and the tree voltages $\underset{\sim}{v}_t$, respectively, via

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

and

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

where $\underset{\sim}{F}$ is the nontrivial part of the fundamental cutset matrix (see Subroutine FCSM), and where the prime denotes the transpose.

Thus we see that Kirchhoff's voltage and current laws are automatically satisfied, for any given network, because $\underset{\sim}{i}_t$ and $\underset{\sim}{v}_\ell$ are forced to satisfy the above relations.

SUBROUTINE INCREM:

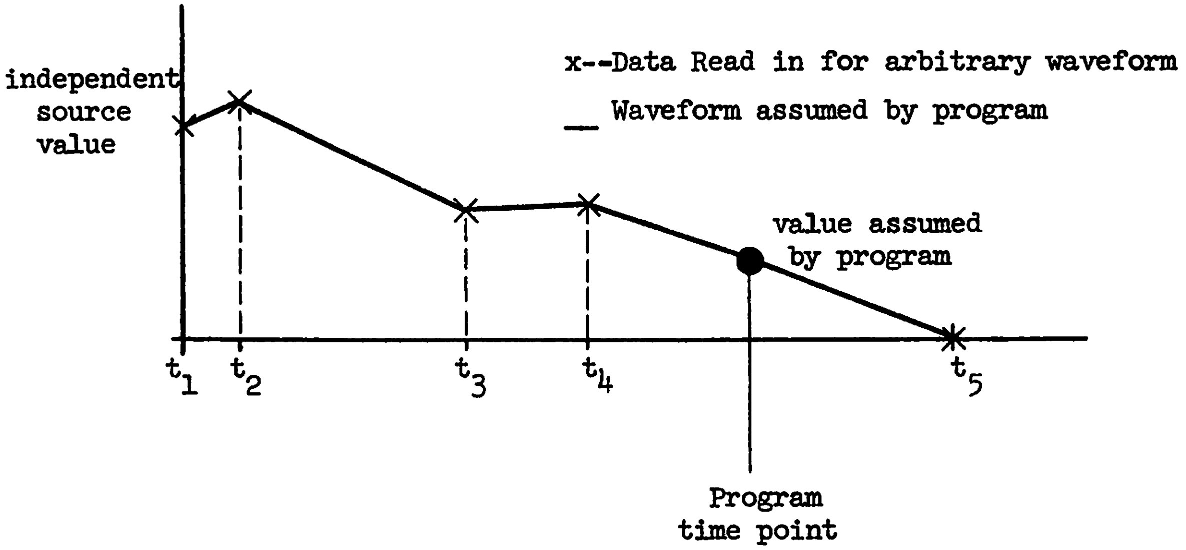
Upon the completion of computations at one time point, this subroutine initializes the program for the next time point.

The most important aspect of the above mentioned initialization, is the computation of the independent source values at the next time point.

When periodic independent sources exist, INCREM calls subroutine PERIOD which then computes the corresponding signal values.

An arbitrary independent source is specified by two arrays (up to 100 points each) denoting the time points and the corresponding source values. If the program time does not correspond to a time point, linear interpolation between the previous and the next time points is automatically carried out.

e.g.



There are three basic ways in which a nonconstant independent source can be specified:

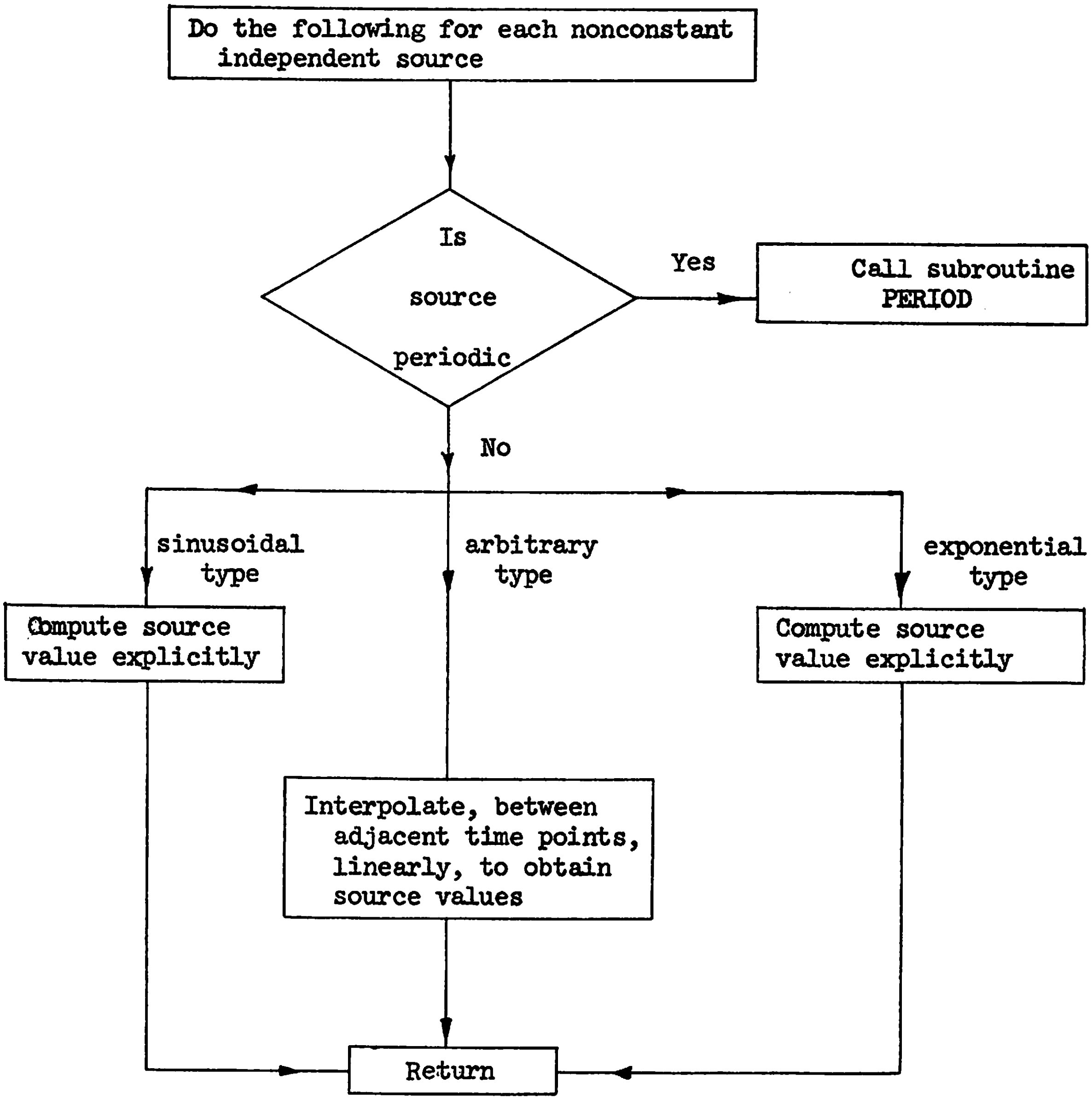
1. arbitrary (as above)
2. exponential
3. sinusoidal

To see how these are read in, see the section on "Data Cards for CANDO." The values of the independent sources, beyond the last specified time point, are taken to be zero.

For sinusoidal and exponential source specifications, the program uses library SINE and EXPONENTIAL routines.

The general operation of this subroutine may be seen on the following page.

BLOCK DIAGRAM


Do the following for each nonconstant independent source
Is source periodic
Yes
Call subroutine PERIOD
No
sinusoidal type
arbitrary type
exponential type
Compute source value explicitly
Compute source value explicitly
Interpolate, between adjacent time points, linearly, to obtain source values
Return

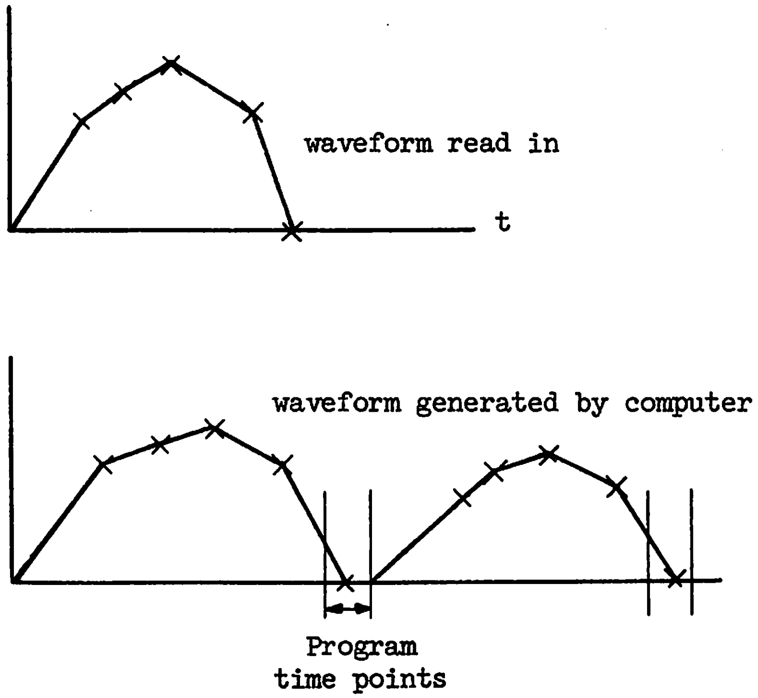
SUBROUTINE INCREM

SUBROUTINE PERIOD:

This subroutine computes the signals of periodic independent sources, for the next time point.

Linear interpolation is used, as in subroutine INCREM, the only difference being that when the last specified time point is exceeded, the waveform is repeated.

e.g.



Thus we note that when the last specified time point does not correspond to a program time point (i.e., $t_0 + nH$ where $n = 0, 1, \ldots$), the period, T, of the waveform is increased so that $T = nH$, $n = 1, 2, \ldots$.

SUBROUTINE ERROR:

This subroutine computes the branch errors E_i and the performance criterion which is to be minimized, defined by

$$\epsilon = \frac{1}{2}\sum_i E_i^2$$

where i ranges over all nonindependent source branches.

Note that when both the branch relations and Kirchhoff's laws are simultaneously satisfied, ϵ will be identically zero.

Branch errors:

Notation: The λ_i denote the branch element constants defined in subroutine KONST. V_{i2} and C_{i2} denote the present branch voltage and current respectively. V_{i1} and C_{i1} denote the branch voltage and current, at the previous time point, respectively. The E_i, of course, denote the branch error associated with the ith branch.

Integration:

The trapezoidal integration scheme is utilized to perform the integrations associated with the reactive element branch relations.

The integral of a function f over an interval H is approximated by

$$\int_{t-H}^{t} f\,dt = \frac{H}{2}\left[f\Big|_{t-H} + f\Big|_{t}\right]$$

Since an iterative minimization scheme is utilized at each time point, the trapezoidal rule, in this context, becomes equivalent to a multistep, second-order predictor corrector scheme.

We always integrate either voltage or current. Since both are treated alike, only the current integration will be considered below.

We have the following typical error term

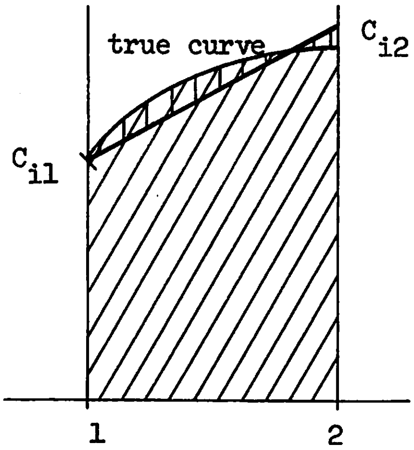
$$E_i = - (V_{i2} - V_{i1}) + \lambda_i * (C_{i2} + C_{i1})$$

where V_{i1} and C_{i1} are known.

As the error is reduced to zero, we have, in effect,

$$V_{i2} - V_{i1} = \lambda_i * (C_{i2} + C_{i1})$$

i.e., the left hand side is the value of the integral, and C_{i2} is chosen so as to satisfy the above equality via linear interpolation.



Note that the area under the linear interpolation is the same as the area under the true curve.

The above argument indicates that for this integration scheme to be accurate, the step size must be much smaller (of the order of 1/10 is sufficient) than the smallest contributing time constant.

<u>Dependent voltage sources</u>

a) voltage controlled case

$$E_i = - V_{i2} + \lambda_i * V_{K2}$$

where the subscript K denotes the controlling branch.

b) current controlled case

$$E_i = - V_{i2} + \lambda_i * C_{K2}$$

where K again denotes the controlling branch.

Capacitances:

a) tree branch case

$$E_i = - (V_{i2} - V_{i1}) + \lambda_i * (C_{i2} + C_{i1})$$

b) link case

$$E_i = - (C_{i2} + C_{i1}) + \lambda_i * (V_{i2} - V_{i1})$$

Note that at the first time point, when we are solving the initial condition problem, E_i is set to zero.

Resistances:

a) tree branch case

$$E_i = - V_{i2} + \lambda_i * C_{i2}$$

b) link case

$$E_i = - C_{i2} + \lambda_i * V_{i2}$$

Inductances:

a) tree branch case

$$E_i = - (V_{i2} + V_{i1}) + \lambda_i * (C_{i2} - C_{i1})$$

b) link case

$$E_i = -(C_{i2} - C_{i1}) + \lambda_i * (V_{i2} + V_{i1})$$

Note that E_i is set equal to zero at the first time point, while solving the initial condition problem.

Dependent current sources:

a) voltage controlled case

$$E_i = - C_{i2} + \lambda_i * V_{K2}$$

where K denotes the controlling branch.

b) current controlled case

$$E_i = - C_{i2} + \lambda_i * C_{K2}$$

where K denotes the controlling branch.

Performance criterion:

The performance criterion ϵ is given by

$$\epsilon = \sum_i E_i^2$$

where i ranges over all the nonindependent source branches.

SUBROUTINE FNDGRAD:

This subroutine computes the gradient of the performance criterion, with respect to nonindependent source tree voltages and nonindependent source link currents.

Theory

Tree voltage gradient vector:

Recall that

$$\underset{\wedge}{v}_b = \begin{bmatrix} \underset{\wedge}{I} \\ --- \\ \underset{\wedge}{F}' \end{bmatrix} \underset{\wedge}{v}_t \tag{1}$$

The tree voltage gradient vector $\frac{d\epsilon}{d\underset{\wedge}{v}_t}$ can be written, via the chain rule, as

$$\frac{d\epsilon}{d\underset{\wedge}{v}_t} = \left[\frac{\partial \underset{\wedge}{v}_b}{\partial \underset{\wedge}{v}_t}\right] \cdot \frac{\partial \epsilon}{\partial \underset{\wedge}{v}_b} \tag{2}$$

where

$$\frac{\partial \underset{\wedge}{v}_b}{\partial \underset{\wedge}{v}_t} = \begin{bmatrix} \frac{\partial v_{b_1}}{\partial v_{t_1}} & \frac{\partial v_{b_2}}{\partial v_{t_1}} & \cdots & \frac{\partial v_{b_{NB}}}{\partial v_{t_1}} \\ \vdots & & \ddots & \vdots \\ \frac{\partial v_{b_1}}{\partial v_{t_{NN-1}}} & \cdots & \cdots & \frac{\partial v_{b_{NB}}}{\partial v_{t_{NN-1}}} \end{bmatrix}$$

and from (1) we have,

$$\frac{\partial \underset{\sim}{v}_b}{\partial \underset{\sim}{v}_t} = \left[\begin{array}{c} \underset{\sim}{I} \\ \hdashline \underset{\sim}{F}' \end{array}\right]' \qquad (3)$$

Note that $\frac{\partial \varepsilon}{\partial \underset{\sim}{v}_b}$ can be partitioned, i.e.,

$$\frac{\partial \varepsilon}{\partial \underset{\sim}{v}_b} = \left[\begin{array}{c} \frac{\partial \varepsilon}{\partial \underset{\sim}{v}_t} \\ \hdashline \frac{\partial \varepsilon}{\partial \underset{\sim}{v}_\ell} \end{array}\right] \qquad (4)$$

Hence, combining (2), (3), and (4) we obtain

$$\frac{d\varepsilon}{d\underset{\sim}{v}_t} = [\underset{\sim}{I} \mid \underset{\sim}{F}] \left[\begin{array}{c} \frac{\partial \varepsilon}{\partial \underset{\sim}{v}_t} \\ \hdashline \frac{\partial \varepsilon}{\partial \underset{\sim}{v}_\ell} \end{array}\right]$$

which reduces to

$$\frac{d\varepsilon}{d\underset{\sim}{v}_t} = \frac{\partial \varepsilon}{\partial \underset{\sim}{v}_t} + \underset{\sim}{F} \cdot \frac{\partial \varepsilon}{\partial \underset{\sim}{v}_\ell} \qquad (5)$$

<u>Link</u> <u>current</u> <u>gradient</u> <u>vector</u>:

Since

$$\underset{\sim}{i}_b = \left[\begin{array}{c} -\underset{\sim}{F} \\ \hdashline \underset{\sim}{I} \end{array}\right] \underset{\sim}{i}_\ell \qquad (6)$$

we may proceed exactly as in the tree voltage gradient case, and arrive at

$$\frac{d\varepsilon}{d\underset{\sim}{i}_{\ell}} = \frac{\partial\varepsilon}{\partial\underset{\sim}{i}_{\ell}} - \underset{\wedge}{F}' \cdot \frac{\partial\varepsilon}{\partial\underset{\wedge}{i}_{t}} \tag{7}$$

Branch Gradients:

The branch gradients $\frac{\partial\varepsilon}{\partial\underset{\wedge}{i}_{b}}$ and $\frac{\partial\varepsilon}{\partial\underset{\wedge}{v}_{b}}$ are computed as follows, where $\frac{\partial\varepsilon}{\partial i_{i}}$ and $\frac{\partial\varepsilon}{\partial v_{i}}$ are branch gradients associated with the ith branch:

Tree branches:

Independent voltage sources:

a) not associated with controlled sources

$$\frac{\partial\varepsilon}{\partial v_{i}} = \frac{\partial\varepsilon}{\partial i_{i}} = 0$$

b) associated with controlled source K

$$\frac{\partial\varepsilon}{\partial v_{i}} = 0$$

$$\frac{\partial\varepsilon}{\partial i_{i}} = \lambda_{K} * E_{K}$$

Dependent voltage sources:

$$\frac{\partial\varepsilon}{\partial v_{i}} = - E_{i}$$

$$\frac{\partial\varepsilon}{\partial i_{i}} = 0$$

Dependent current sources:

$$\frac{\partial \varepsilon}{\partial v_i} = 0$$

$$\frac{\partial \varepsilon}{\partial i_i} = - E_i$$

Remaining tree branches:

$$\frac{\partial \varepsilon}{\partial v_i} = - E_i$$

$$\frac{\partial \varepsilon}{\partial i_i} = \lambda_i * E_i$$

Link branches:

Independent current sources:

a) not associated with controlled sources

$$\frac{\partial \varepsilon}{\partial v_i} = \frac{\partial \varepsilon}{\partial i_i} = 0$$

b) associated with controlled source K

$$\frac{\partial \varepsilon}{\partial v_i} = \lambda_K * E_K$$

$$\frac{\partial \varepsilon}{\partial i_i} = 0$$

Dependent current sources:

$$\frac{\partial \varepsilon}{\partial v_i} = 0$$

$$\frac{\partial \varepsilon}{\partial i_i} = - E_i$$

Dependent voltage sources:

$$\frac{\partial \varepsilon}{\partial v_i} = - E_i$$

$$\frac{\partial \varepsilon}{\partial i_i} = 0$$

Remaining link branches:

$$\frac{\partial \varepsilon}{\partial v_i} = \lambda_i * E_i$$

$$\frac{\partial \varepsilon}{\partial i_i} = - E_i$$

Note, when solving the initial condition problem, the gradients with respect to reactive element currents and/or voltages, are set equal to zero.

SUBROUTINE FLPOW:

This subroutine computes the direction $\underset{\wedge}{S}$ along which we wish to minimize the performance criterion ε.

The Fletcher-Powell(1) algorithm is utilized with the initial direction being that of steepest descent (i.e., along the gradient).

Fletcher-Powell Algorithm:

Let $\underset{\wedge}{H}$ be a positive definite symmetric matrix.

The direction along which we perform the minimization is given by

$$\underset{\wedge}{S}^{i} = \underset{\wedge}{H}^{i} \cdot \underset{\wedge}{g}^{i}$$

where the superscript i indicates that we are at the ith iterative step, and $\underset{\wedge}{g}$ is the gradient vector.

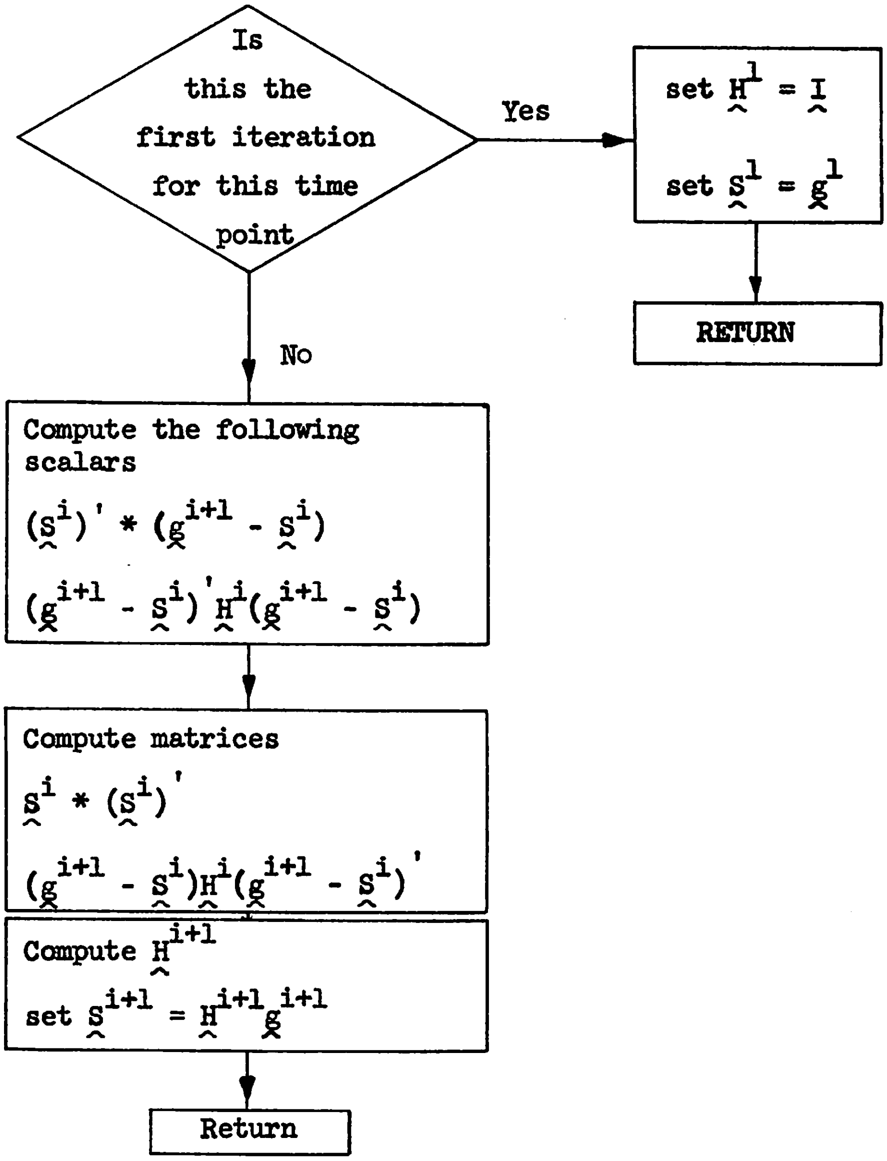
$\underset{\wedge}{H}$ is updated as follows

$$\underset{\wedge}{H}^{1} = \underset{\wedge}{I}$$

$$\underset{\wedge}{H}^{i+1} = \underset{\wedge}{H}^{i} - \frac{\underset{\wedge}{S}^{i} * (\underset{\wedge}{S}^{i})'}{(\underset{\wedge}{S}^{i})' * (\underset{\wedge}{g}^{i+1} - \underset{\wedge}{S}^{i})} - \frac{(\underset{\wedge}{g}^{i+1} - \underset{\wedge}{S}^{i})\underset{\wedge}{H}^{i}(\underset{\wedge}{g}^{i+1} - \underset{\wedge}{S}^{i})'}{(\underset{\wedge}{g}^{i+1} - \underset{\wedge}{S}^{i})'\underset{\wedge}{H}^{i}(\underset{\wedge}{g}^{i+1} - \underset{\wedge}{S}^{i})}$$

Thus we see that use of the Fletcher-Powell algorithm requires us to store an n × n matrix, the Hessian matrix.

Block Diagram:


Is this the first iteration for this time point
Yes
set $\underset{\wedge}{H}^1 = \underset{\wedge}{I}$
set $\underset{\wedge}{S}^1 = \underset{\approx}{g}^1$
RETURN
No
Compute the following scalars
$(\underset{\wedge}{S}^i)' * (\underset{\approx}{g}^{i+1} - \underset{\wedge}{S}^i)$
$(\underset{\approx}{g}^{i+1} - \underset{\wedge}{S}^i)' \underset{\wedge}{H}^i (\underset{\approx}{g}^{i+1} - \underset{\wedge}{S}^i)$
Compute matrices
$\underset{\wedge}{S}^i * (\underset{\wedge}{S}^i)'$
$(\underset{\approx}{g}^{i+1} - \underset{\wedge}{S}^i) \underset{\wedge}{H}^i (\underset{\approx}{g}^{i+1} - \underset{\wedge}{S}^i)'$
Compute $\underset{\wedge}{H}^{i+1}$
set $\underset{\wedge}{S}^{i+1} = \underset{\wedge}{H}^{i+1} \underset{\approx}{g}^{i+1}$
Return

SUBROUTINE FLPOW

SUBROUTINE FNDALPH

This subroutine computes α^i, the optimal step size at the ith iterative step, which minimizes ε along $\underset{\wedge}{S}^i$, a direction computed by subroutine FLPOW.

The constant α^i is such that

$$\min_{\lambda} \varepsilon(\underset{\wedge}{x}^i - \lambda \underset{\wedge}{S}^i) = \varepsilon(\underset{\wedge}{x}^i - \alpha^i \underset{\wedge}{S}^i)$$

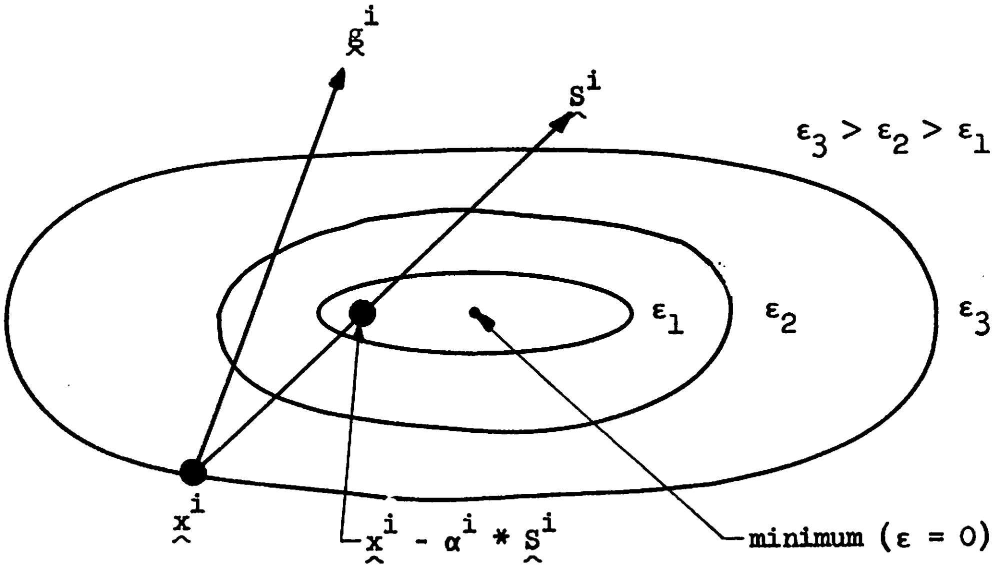
where $\underset{\wedge}{x}$ is the set of nonindependent source tree voltages and link currents, at the ith iterative step.

The analytical expression for α^i is:

$$\alpha^i = \frac{1}{2} \frac{(\underset{\wedge}{g}^i)' * \underset{\wedge}{S}^i}{\varepsilon(\underset{\wedge}{x} - \underset{\wedge}{S}^i) + (\underset{\wedge}{g}^i)' * \underset{\wedge}{S}^i - \varepsilon(\underset{\wedge}{x})}$$

where $\underset{\wedge}{g}^i$ is the gradient at the ith iterative step.

The situation may be depicted graphically as follows:



$\underset{\wedge}{g}^i$
$\underset{\wedge}{S}^i$
$\varepsilon_3 > \varepsilon_2 > \varepsilon_1$
ε_1
ε_2
ε_3
$\underset{\wedge}{x}^i$
$\underset{\wedge}{x}^i - \alpha^i * \underset{\wedge}{S}^i$
minimum (ε = 0)

SUBROUTINE CNGVARS:

Given the direction along which we minimized, $\underset{\frown}{S}^i$, and the optimal step size α^i, the nonindependent source tree voltages and link currents are modified as follows:

$$\begin{bmatrix} \underset{\frown}{v}_t \\ --- \\ \underset{\frown}{i}_\ell \end{bmatrix}_{\text{new}} = \begin{bmatrix} \underset{\frown}{v}_t \\ --- \\ \underset{\frown}{i}_\ell \end{bmatrix}_{\text{old}} - \alpha^i * \underset{\frown}{S}^i$$

It should be emphasized that only the basis set of variables, i.e., tree voltages and link currents, are changed. Tree currents and link voltages follow directly from the implicit formulation of Kirchhoff's laws.

SUBROUTINE READOUT:

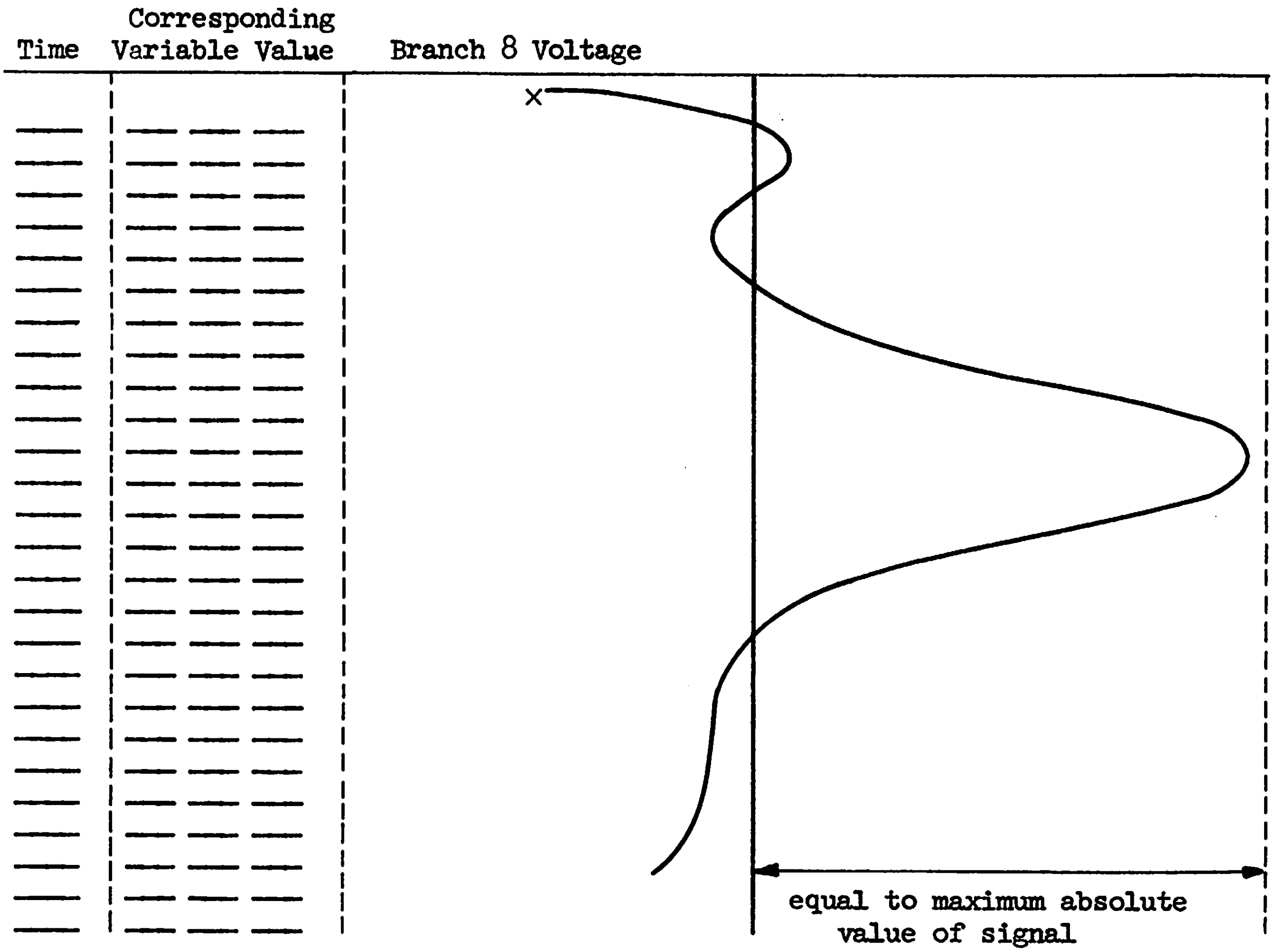
This subroutine outputs all the desired currents and/or voltages, in the desired order, and in the original network numbering scheme. Hence for any branch, one may output its current and/or its voltage.

Note that a one to one correspondence, between desired output variables and their internal ordering, needs to be calculated each time a new tree is selected. This is done in subroutine PT.

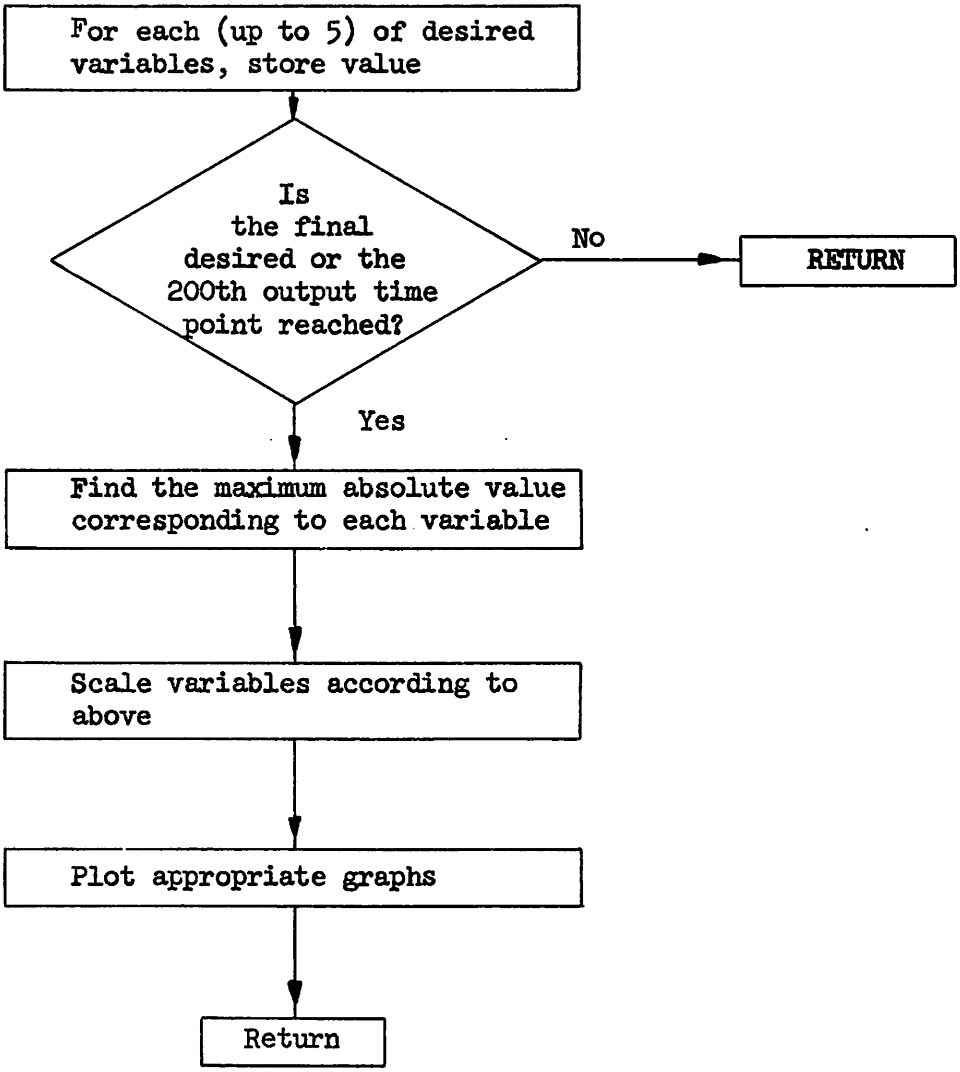
<u>SUBROUTINE</u> <u>GRAPH</u>: (Graphical outputs)

The purpose of this subroutine is to provide graphical outputs of up to five variables. For each time point at which an output is desired, the main program calls this subroutine. The desired variables are then stored in an array. When either the final desired output time point or the two hundredth output time point is reached, the variables will be scaled, individually, such that maximum use will be made of the output page width. The corresponding variable value is also outputed.

The following figure illustrates a typical graph output.


Time
Corresponding Variable Value
Branch 8 Voltage
x
equal to maximum absolute value of signal

The operation of this subroutine may be seen on the following page.


For each (up to 5) of desired variables, store value
Is the final desired or the 200th output time point reached?
No
RETURN
Yes
Find the maximum absolute value corresponding to each variable
Scale variables according to above
Plot appropriate graphs
Return

SUBROUTINE GRAPH

SUBROUTINE ALLOUT:

This subroutine outputs all the tree voltages and link currents (including those associated with independent sources), in the optimal or proper tree numbering scheme. Hence, when it is used, one should also output the corresponding tree information which will immediately yield the isomorphism between the original network and the new topologies.

Table of Contents

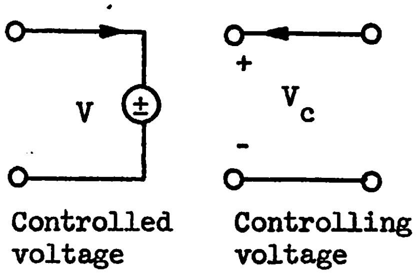
APPENDIX A:

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.

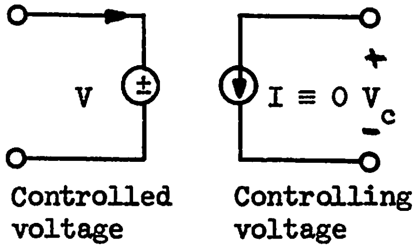
Voltage controlled voltage source:

Ideal model:



$V \equiv kV_c$

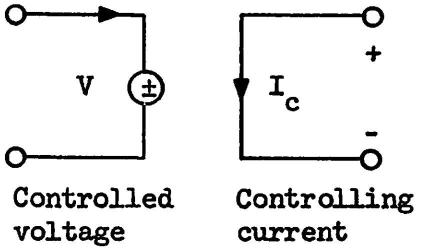
CANDO model:



$V \equiv kV_c$

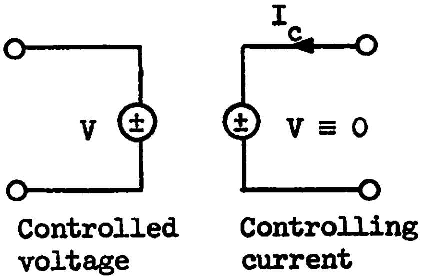
Current controlled voltage source:

Ideal model:



$V \equiv kI_c$

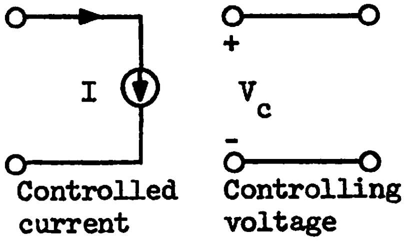
CANDO model:



$V \equiv kI_c$

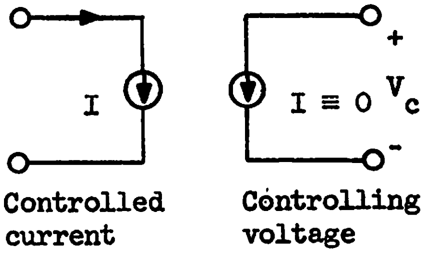
Voltage controlled current source:

Ideal model:



$I \equiv kV_c$

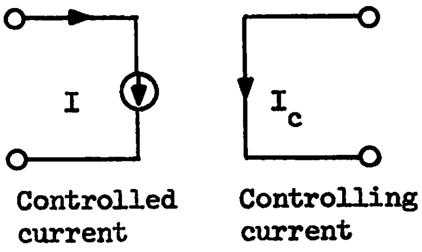
CANDO model:



$I \equiv kV_c$

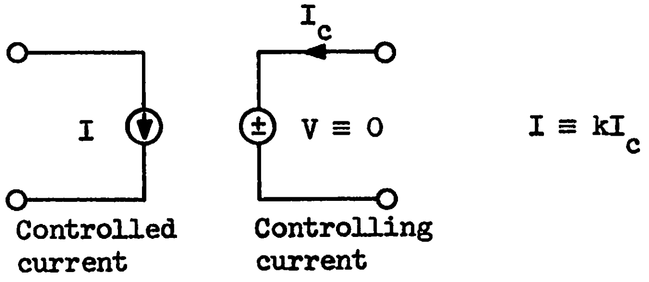
Current controlled current source:

Ideal model:



$I \equiv kI_c$

CANDO model:



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 CANDO

Notation:

I ⇒ Integer format

E ⇒ Exponential or floating point format

A ⇒ Alphanumeric format

Col ⇒ 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)

EPS, error criterion (E)

NCONT, Tree and $\underset{\wedge}{F}$ output control (I)

NSTEP = 1 ⇒ outputs desired

NSTEP = 0 ⇒ outputs not desired

col 1-5,	-	NN
col 6-10,	-	NB
col 11-25,	-	TSTART
col 26-40,	-	TEND
col 41-55,	-	H
col 56-60,	-	NSTEP
col 61-75,	-	EPS
col 80,	-	NCONT

DATA CARD #1

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 ⇒ use of ALLOUT is desired

NALLOUT = 0 ⇒ use of ALLOUT is not desired

JOUT, number of outputs desired (I)

To be used in conjunction with subroutine READOUT.

SCALE, scale factor (E)

e.g., scale factor of 10^3 sets current unit to milliamps

NITT, number of iterative minimization steps, per time point (I)

(NITT = 0 if using only error criterion)

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

NSCALE = 1 ⇒ use scale factor read in as SCALE

NSCALE ≠ 1 ⇒ use internal, automatic current scaling algorithm.

col 1-5,	-	NGRAPH	DATA CARD #2
col 6-10,	-	NALLOUT	
col 11-15,	-	JOUT	
col 16-30,	-	SCALE	
col 31-35,	-	NITT	
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 ⇒ no more data needed

NCARDS ≠ 0 ⇒ 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.

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	DATA CARD FOLLOWED BY OTHER RELATED CARDS IF NCARDS ≠ 0
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	

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

- If SORTYPE = S, i.e., a sinusoidal source, the signal is assumed to be of the form

$$A \sin (\omega * t - \Phi)$$

<u>one</u> card is required to describe the above

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

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

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

- If SORTYPE = E, i.e., an exponential source, the signal is assumed to be of the form

$$Ae^{\gamma(t-\Phi)}$$

one card is required to describe the above

col 1-10, signal amplitude A, 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, i.e., 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 allotted 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 allotted 10 columns of space (E).

<u>OUTPUT</u> <u>specifications</u>:

Only one type of output is allowed for any one network analysis, i.e., 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) 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) 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:

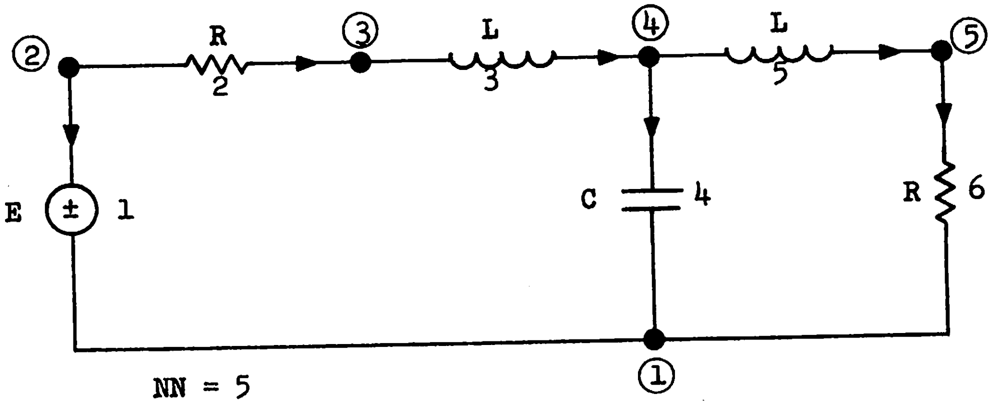
Notation:

- 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
- EPS desired performance criterion error

Note, that only the pertinent computer output is included with the sample problems.

Sample Problem #1:

Butterworth Filter:



$R = 10^3$

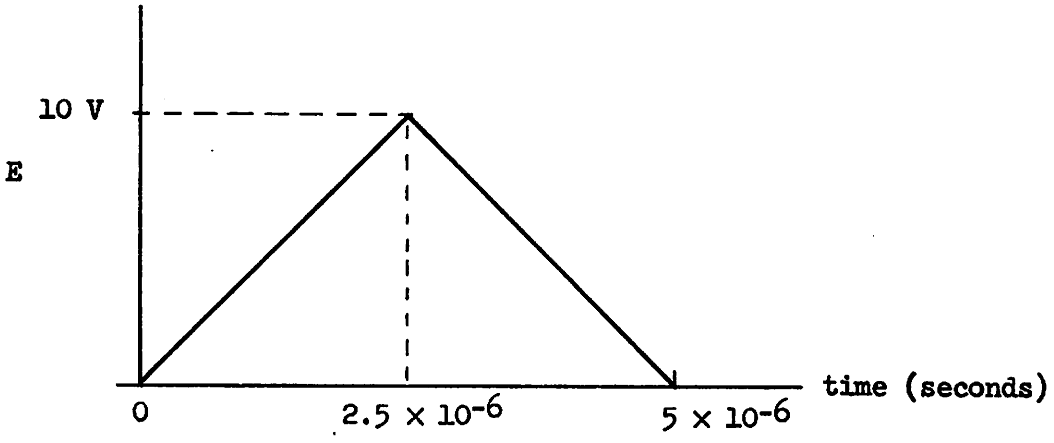
$L = 0.01$

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

NN = 5

NB = 6

The input E is specified to be



<u>Specifications</u>:

- Initial time = 0 (TSTART = 0.0)
- Final time = 10^{-4} (TEND = 0.0001)
- Time increment = 2×10^{-6} (H = 0.000002)
- Desire error, at each time point, to fall below 10^{-7} (EPS = 10^{-7})
- Desire results, at each time point, to be outputed (NSTEP = 1)
- Desire tree and $\underset{\wedge}{F}$ to be outputed (NCONT = 1)
- Desire current to be in milliamps (SCALE = 10^3 and NSCALE = 1)
- Desire the following <u>graphical</u> outputs

$$\left.\begin{array}{ll} \text{Branch 1} & \text{voltage} \\ \text{Branch 2} & \text{current} \\ \text{Branch 6} & \text{voltage} \end{array}\right\} \quad \text{(NGRAPH = 3)}$$

- Specify zero initial conditions.

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

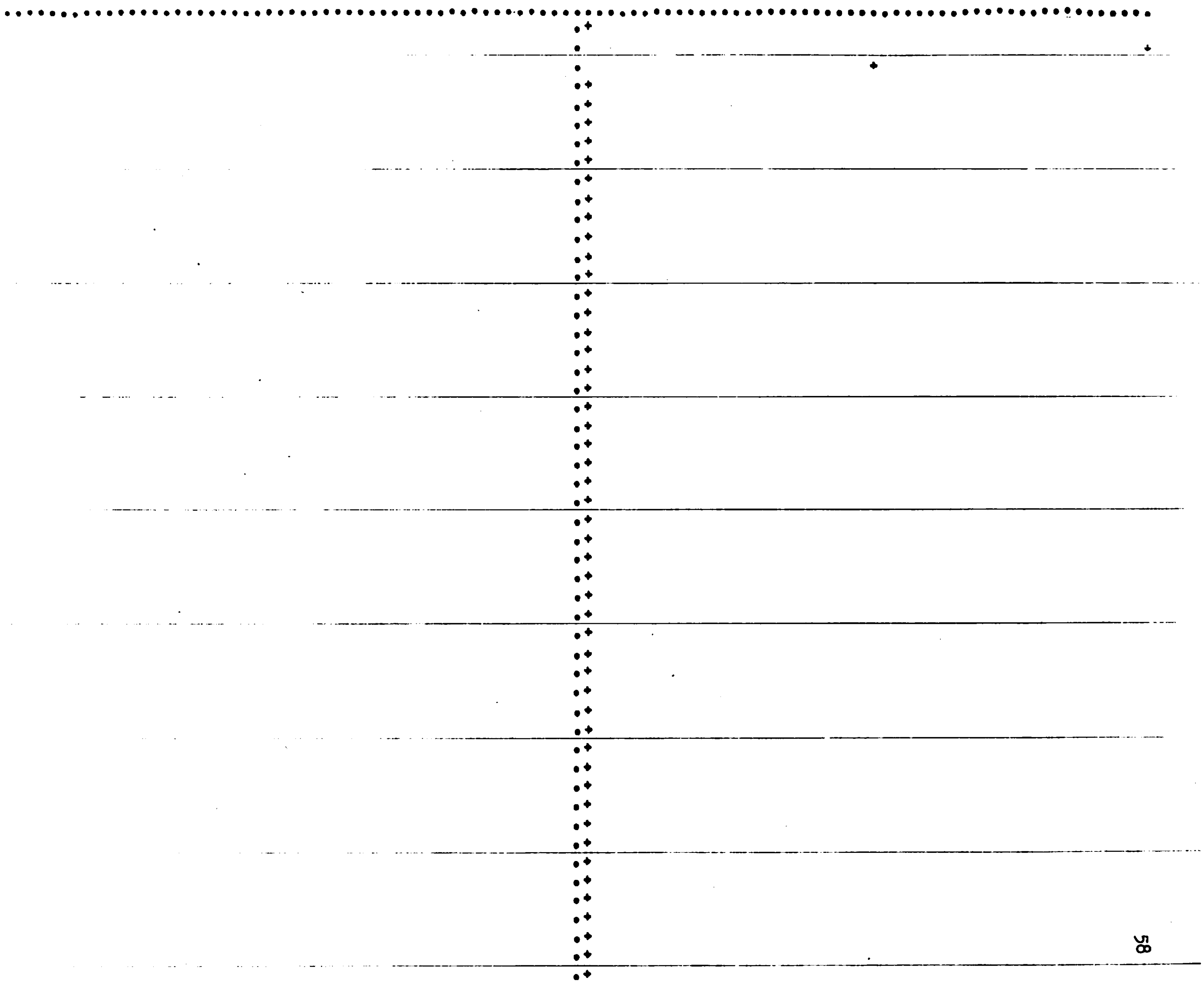
The central processor time for this problem was 2.954 seconds.

DATA CARDS : PROBLEM # 1

```
    5    6            0.0         0.0001        0.2E-05    1       0.1E-06
    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
```

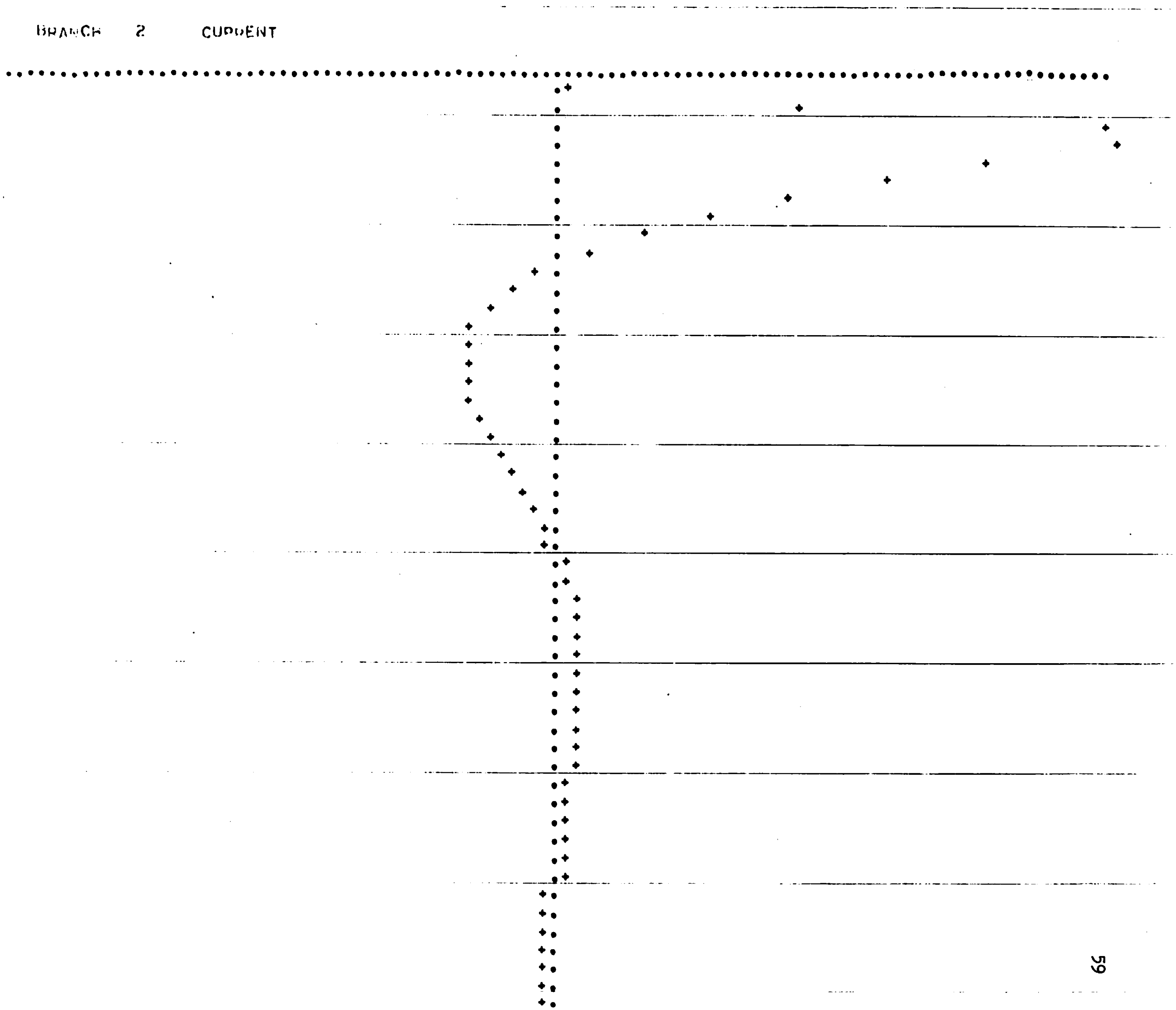
```
                               BRANCH   1      VOLTAGE

 TIME           VALUE
0.              0.
.2000E-05       .2000E+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.
```



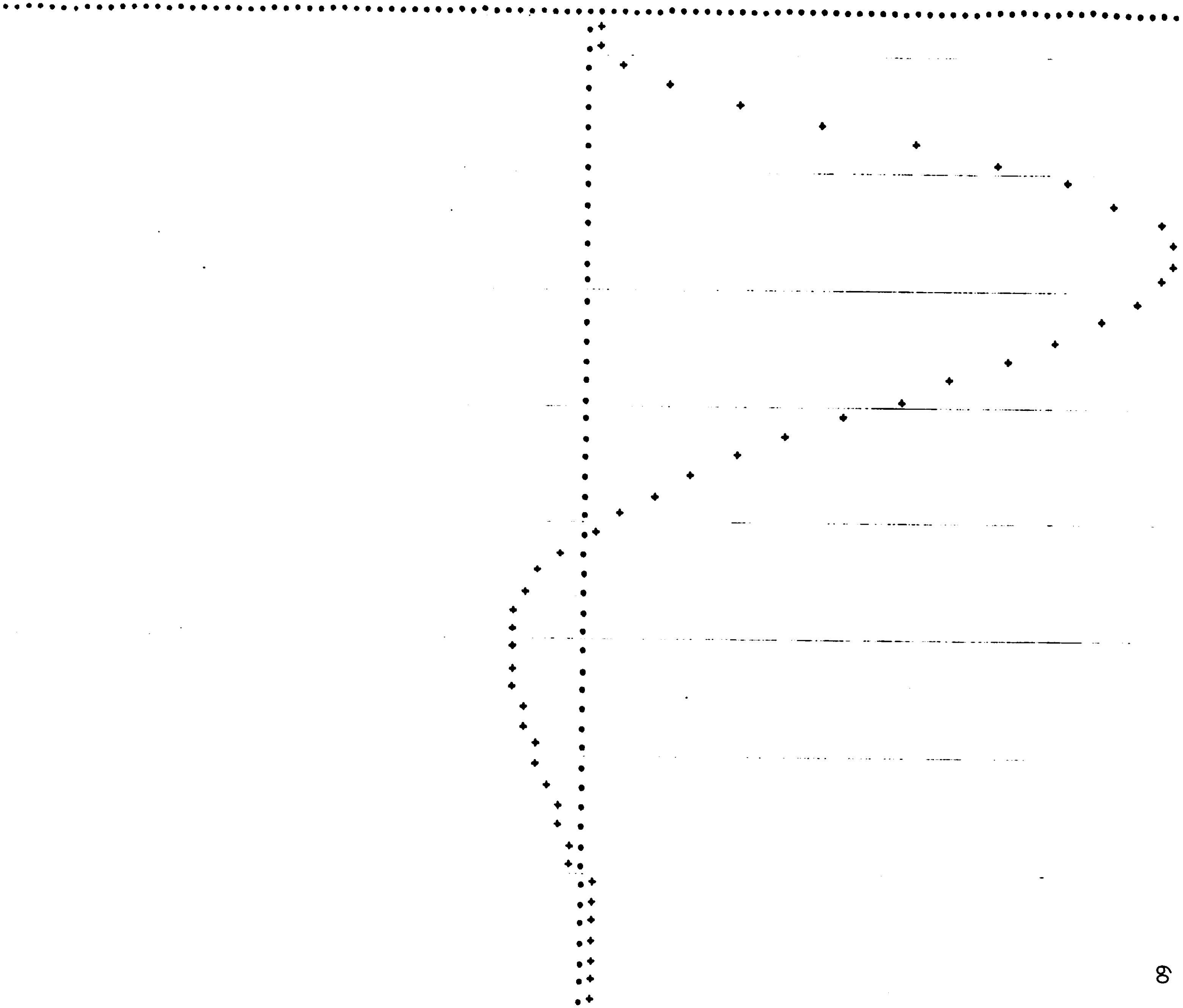
```
BRANCH  2      CURRENT

 TIME          VALUE
 0.            0.
.2000E-05     .7241E-03
.4000E-05     .1666E-02
.6000E-05     .1684E-02
.8000E-05     .1367E-02
.1000E-04     .9771E-03
.1200E-04     .6926E-03
.1400E-04     .4518E-03
.1600E-04     .2523E-03
.1800E-04     .9135E-04
.2000E-04    -.3442E-04
.2200E-04    -.1286E-03
.2400E-04    -.1948E-03
.2600E-04    -.2369E-03
.2800E-04    -.2588E-03
.3000E-04    -.2638E-03
.3200E-04    -.2555E-03
.3400E-04    -.2373E-03
.3600E-04    -.2120E-03
.3800E-04    -.1820E-03
.4000E-04    -.1495E-03
.4200E-04    -.1155E-03
.4400E-04    -.8438E-04
.4600E-04    -.5435E-04
.4800E-04    -.2716E-04
.5000E-04    -.3415E-05
.5200E-04     .1655E-04
.5400E-04     .3259E-04
.5600E-04     .4482E-04
.5800E-04     .5348E-04
.6000E-04     .5889E-04
.6200E-04     .6135E-04
.6400E-04     .6144E-04
.6600E-04     .5921E-04
.6800E-04     .5533E-04
.7000E-04     .5021E-04
.7200E-04     .4430E-04
.7400E-04     .3794E-04
.7600E-04     .3145E-04
.7800E-04     .2505E-04
.8000E-04     .1894E-04
.8200E-04     .1327E-04
.8400E-04     .8158E-05
.8600E-04     .3763E-05
.8800E-04    -.1394E-06
.9000E-04    -.3238E-05
.9200E-04    -.5652E-05
.9400E-04    -.7427E-05
.9600E-04    -.8620E-05
.9800E-04    -.9109E-05
.1000E-03    -.9360E-05
```



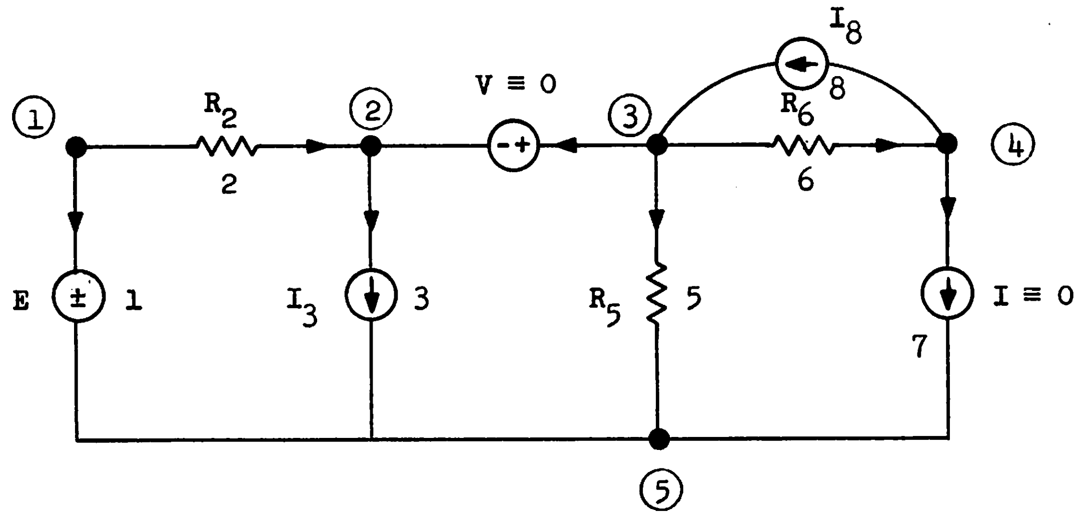
BRANCH 6 VOLTAGE

```
 TIME          VALUE
0.            -0.
 .2000E-05     .3957E-02
 .4000E-05     .2213E-01
 .6000E-05     .5929E-01
 .8000E-05     .1194E+00
 .1000E-04     .1[illegible]E+00
 .1200E-04     .2417E+00
 .1400E-04     .3249E+00
 .1600E-04     .3863E+00
 .1800E-04     .4310E+00
 .2000E-04     .4616E+00
 .2200E-04     .4781E+00
 .2400E-04     .4806E+00
 .2600E-04     .4795E+00
 .2800E-04     .4500E+00
 .3000E-04     .4202E+00
 .3200E-04     .3835E+00
 .3400E-04     .3419E+00
 .3600E-04     .2974E+00
 .3800E-04     .2518E+00
 .4000E-04     .2066E+00
 .4200E-04     .1631E+00
 .4400E-04     .1224E+00
 .4600E-04     .8564E-01
 .4800E-04     .5270E-01
 .5000E-04     .2428E-01
 .5200E-04     .5549E-03
 .5400E-04    -.1852E-01
 .5600E-04    -.3322E-01
 .5800E-04    -.4393E-01
 .6000E-04    -.5107E-01
 .6200E-04    -.5503E-01
 .6400E-04    -.5584E-01
 .6600E-04    -.5473E-01
 .6800E-04    -.5179E-01
 .7000E-04    -.4732E-01
 .7200E-04    -.4199E-01
 .7400E-04    -.3608E-01
 .7600E-04    -.2993E-01
 .7800E-04    -.2379E-01
 .8000E-04    -.1799E-01
 .8200E-04    -.1240E-01
 .8400E-04    -.7431E-02
 .8600E-04    -.3125E-02
 .8800E-04     .6392E-03
 .9000E-04     .3635E-02
 .9200E-04     .5966E-02
 .9400E-04     .7475E-02
 .9600E-04     .8822E-02
 .9800E-04     .9216E-02
 .1000E-03     .9339E-02
```



Sample Problem #2:

Controlled Source DC problem:



$E = 1$ volt DC

$R_2 = R_5 = R_6 = 1\ \Omega$

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 we have a resistive DC problem, we need the solution at only one time point

  set TSTART = 0.0 = TEND

- desire error to fall below 10^{-3}

  ($EPS = 10^{-3}$)

- the time increment = 0

  (set H = 0.0)

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

- do not desire tree and $\underset{\wedge}{F}$ 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 0.24 seconds.

DATA CARDS : PROBLEM # 2

```
    5    8            0.0            0.0            0.0    1          0.001
             16           1.0         1
E  1C      1  5                                1.0
R  2       1  2                 1.0
I   3I  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
```

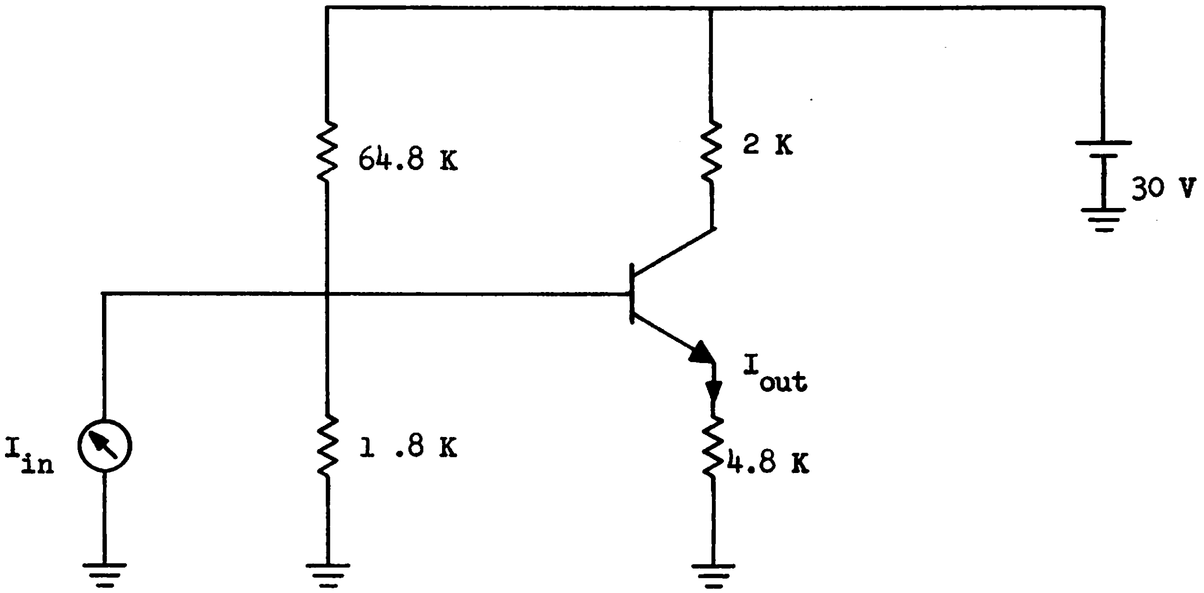
PROBLEM #2 CONT'D

```
    7    0
    8    1
    8    0
```

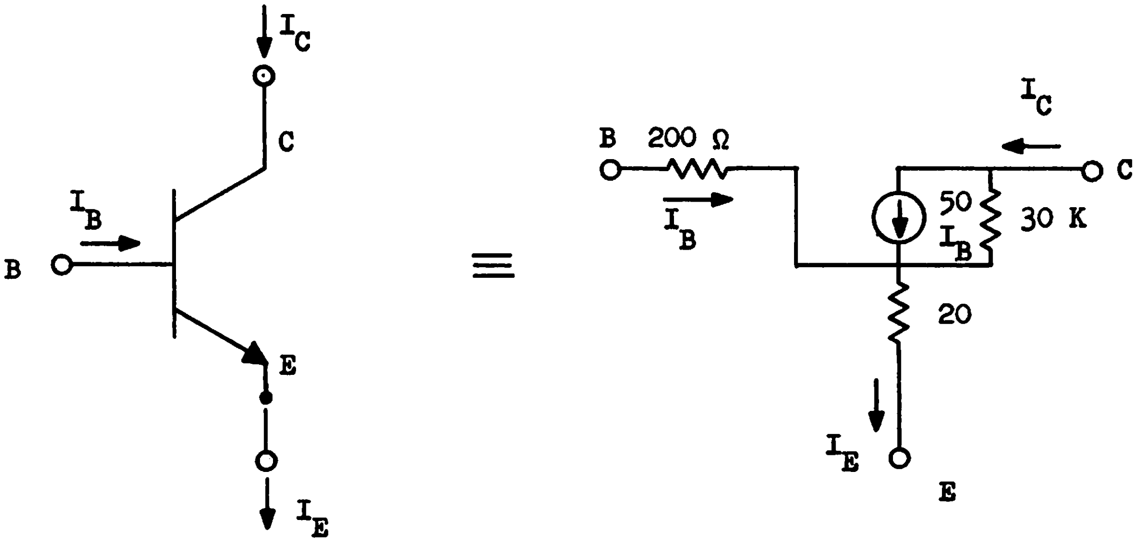
```
TIME =      0.
 BRANCH     1      CURRENT                  -.13173E+01
 BRANCH     1      VOLTAGE                   .10000E+01
 BRANCH     2      CURRENT                   .13173E+01
 BRANCH     2      VOLTAGE                   .13089E+01
 BRANCH     3      CURRENT                   .16420E+01
 BRANCH     3      VOLTAGE                  -.30888E+00
 BRANCH     4      CURRENT                   .32472E+00
 BRANCH     4      VOLTAGE                -0.
 BRANCH     5      CURRENT                  -.32472E+00
 BRANCH     5      VOLTAGE                  -.30888E+00
 BRANCH     6      CURRENT                  -.23257E+00
 BRANCH     6      VOLTAGE                  -.23055E+00
 BRANCH     7      CURRENT                 C.
 BRANCH     7      VOLTAGE                  -.78332E-01
 BRANCH     8      CURRENT                  -.23257E+00
 BRANCH     8      VOLTAGE                   .23055E+00
```

Sample Problem #3:

D. C. amplifier:

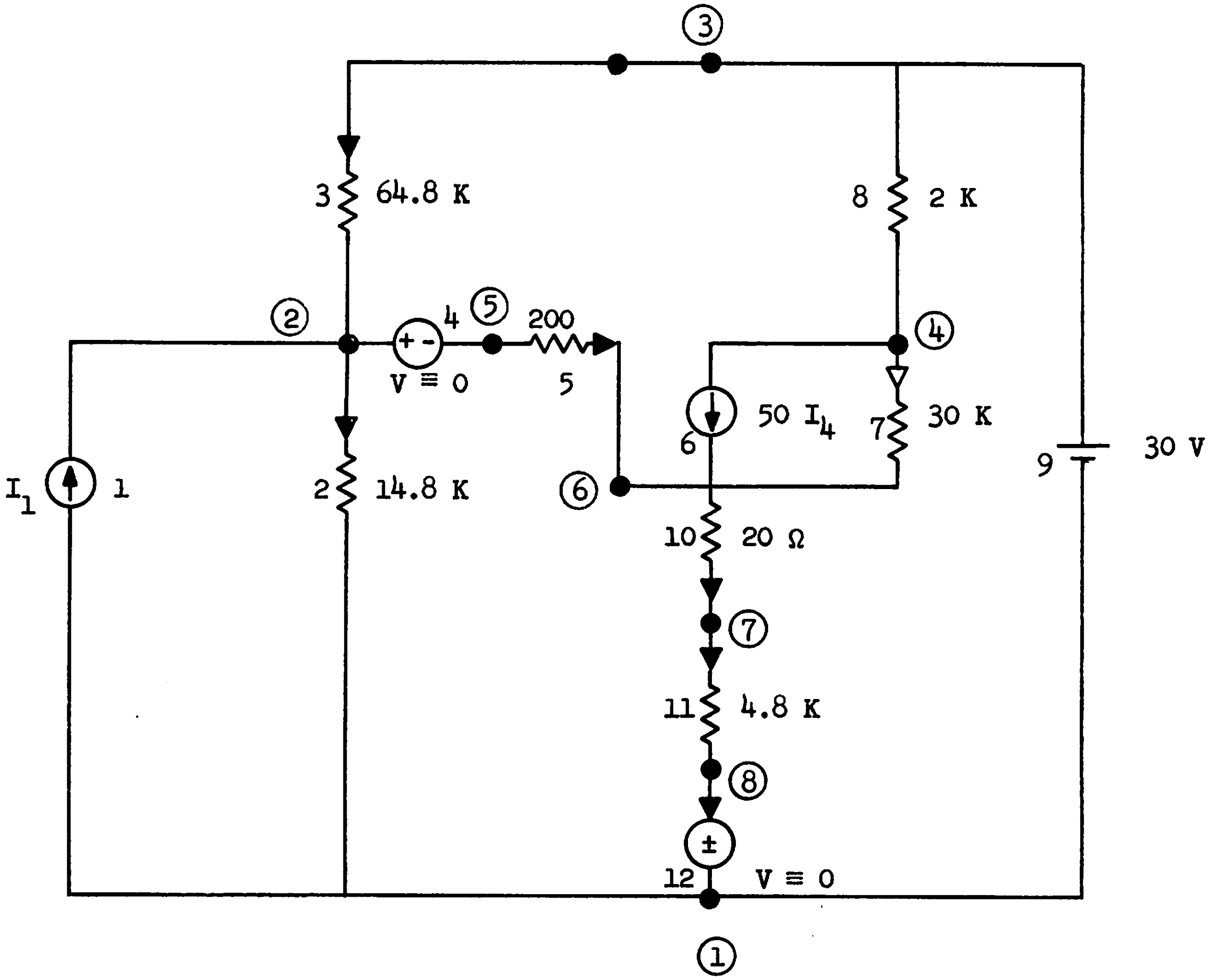


Transistor model:



The output will be the emitter current, and will be taken to be a current through a zero valued voltage source.

Modeling the controlled source in the way described in the section on "Dependent Source Modeling," our network becomes



__Approximate__ __theoretical__ __gain:__

Neglecting the 30K resistor, we can show that $\frac{\Delta I_{12}}{\Delta I_1} \triangleq 2.4$, about a quiescent I_{12} of 1.1 ma.

Specifications:

- I_1 = A sin (1.8t)

  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 error, at each time point, to drop to 10^{-6} or desire to proceed to next time point after 30 iterations (EPS = 10^{-6} and NITT = 30)
- desire tree and $\underset{\wedge}{F}$ to be outputed (NCONT = 1)
- desire results, at each time point, to be outputed (NSTEP = 1)
- desire internal, automatic current scaling (NSCALE ≠ 1)
- desire following graphical outputs

$$\left.\begin{array}{l}\text{Branch 1 current}\\ \text{Branch 4 current}\\ \text{Branch 12 current}\\ \text{Branch 6 current}\end{array}\right\}\qquad \text{(NGRAPH = 4)}$$

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

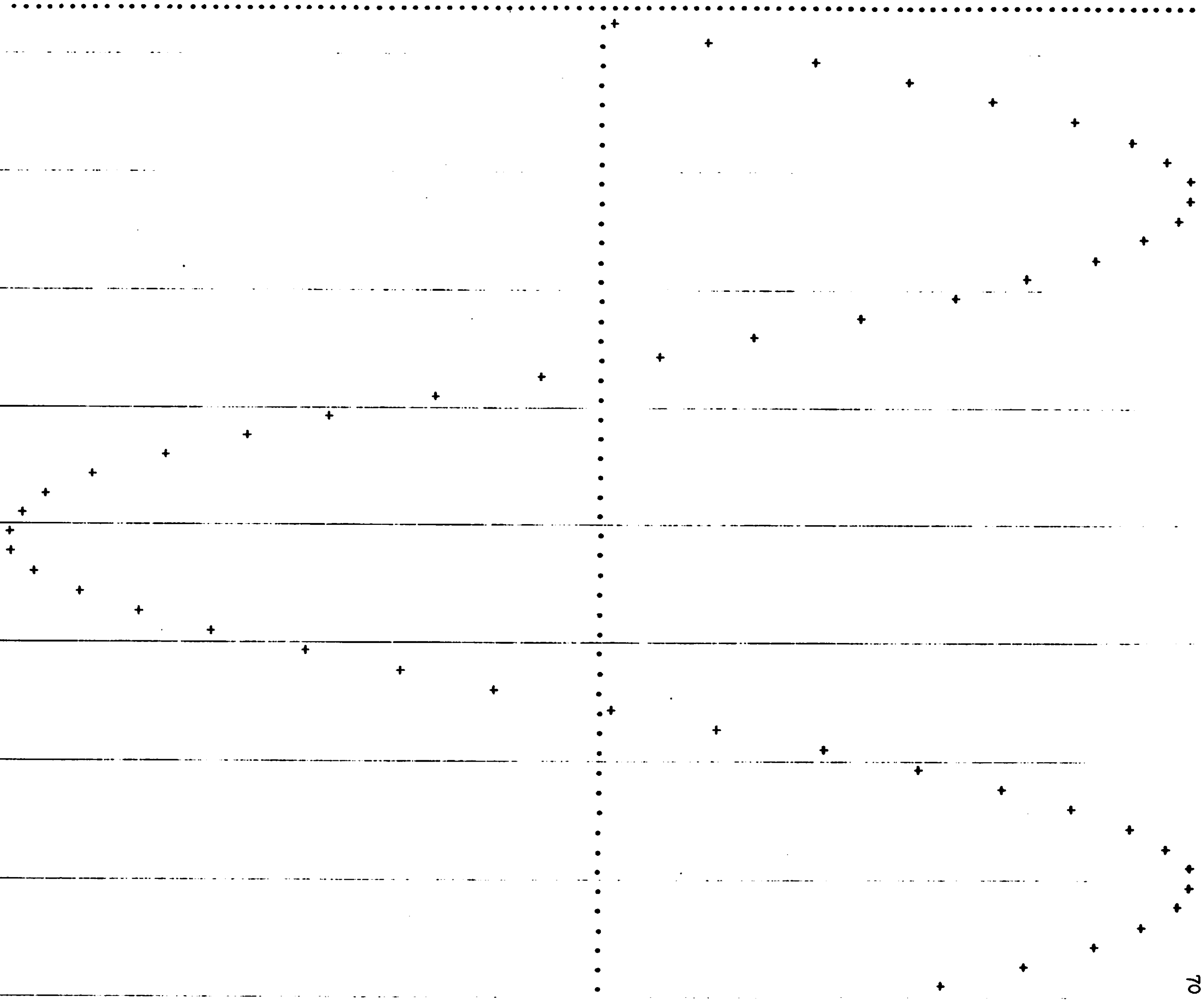
The central processor time, for this problem, was 8.24 seconds.

DATA CARDS : PROBLEM #3

```
    8   12            0.0            5.0            0.1    1        1.0E-06
    4                            30    0
J  1S      1  2  1
    0.0001       1.8       0.0
E  4C      2  5
R  5       5  6               200.0
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
```

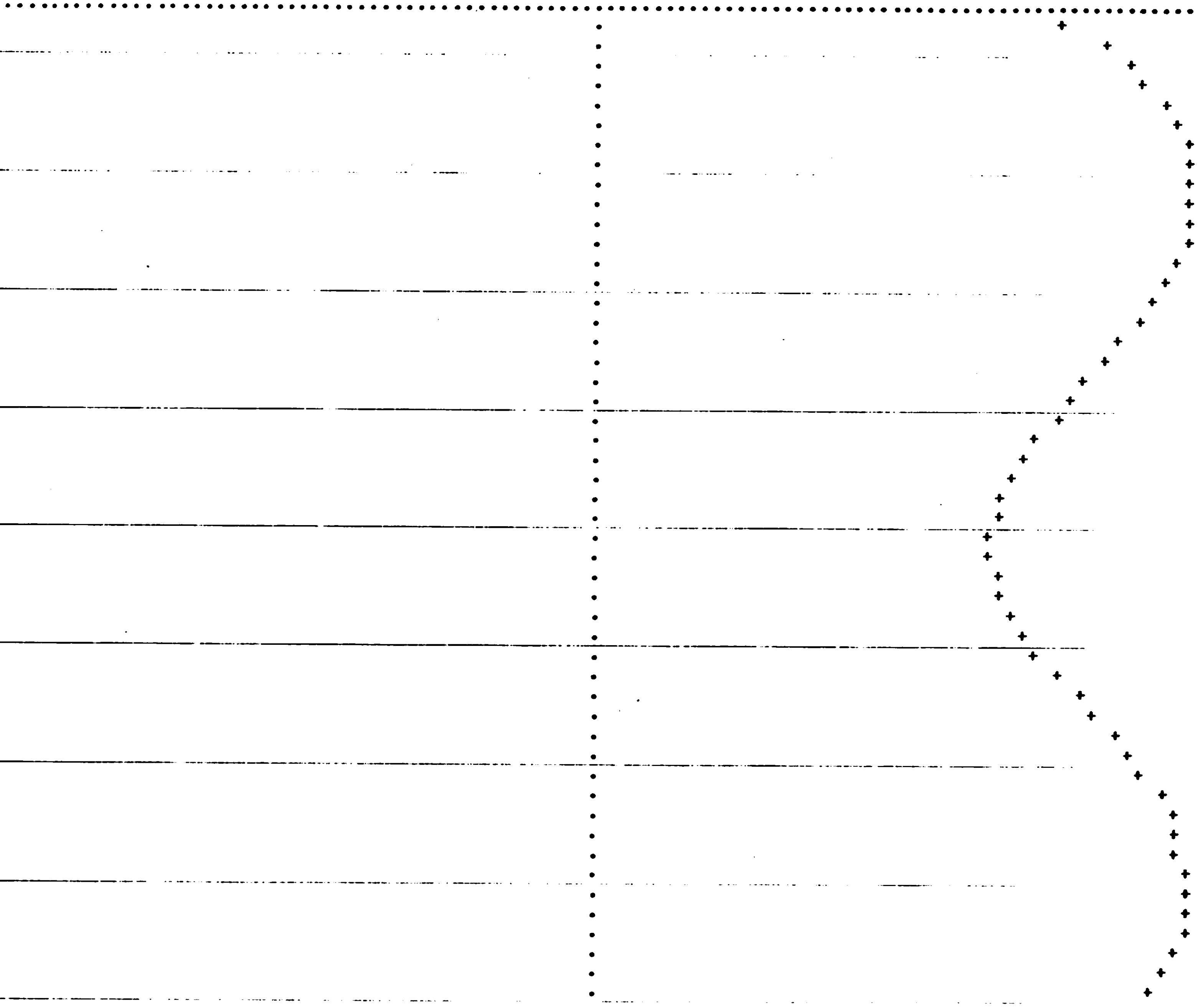
BRANCH 1 CURRENT

```
  TIME          VALUE
 0.            0.
  .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     .9983E-04
  .1000E+01     .9733E-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     .8170E-05
 .1800E+01    -.9828E-05
 .1900E+01    -.2749E-04
 .2000E+01    -.4425E-04
 .2100E+01    -.5953E-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     .1481E-05
 .3600E+01     .1955E-04
 .3700E+01     .3680E-04
 .3800E+01     .5288E-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     .5684E-04
```

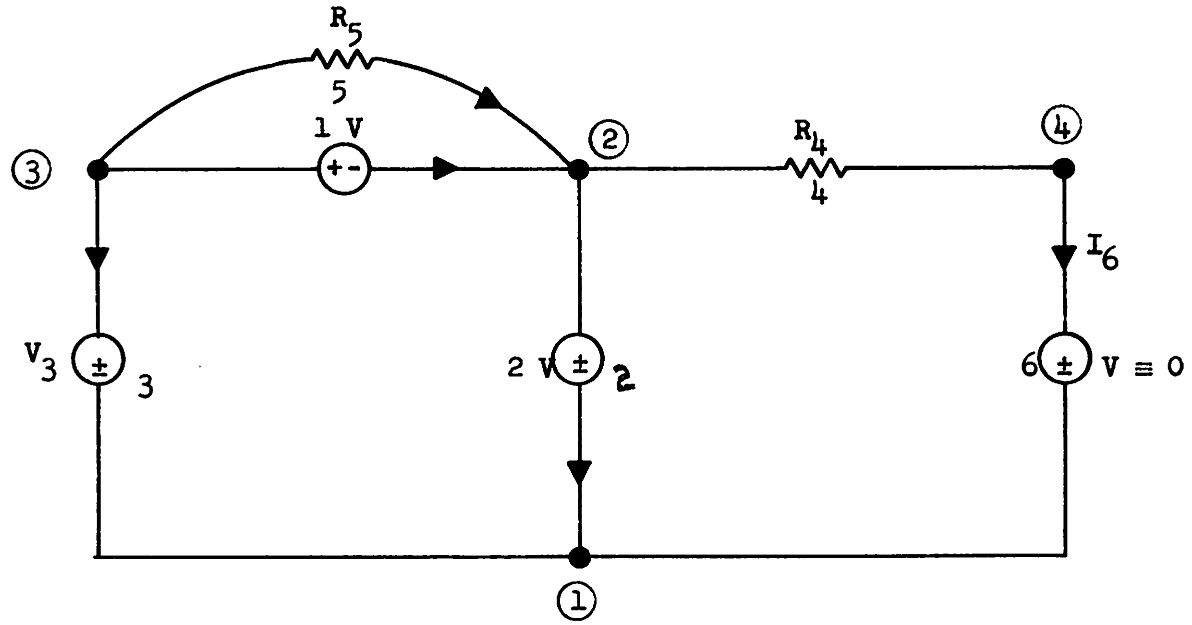


```
                         BRANCH  12        CURRENT

    TIME          VALUE
 0.              .1054 -02
  .10000+00      .1190 -02
  .20000+00      .1220 -02
  .30000+00      .1259 -02
  .40000+00      .1293 -02
  .50000+00      .1322 -02
  .60000+00      .1345 -02
  .70000+00      .1361 -02
  .80000+00      .1371 -02
  .90000+00      .1372 -02
  .10000+01      .1367 -02
  .11000+01      .1354 -02
  .12000+01      .1334 -02
  .13000+01      .1307 -02
  .14000+01      .1275 -02
  .15000+01      .1238 -02
  .16000+01      .1199 -02
  .17000+01      .1157 -02
  .18000+01      .1114 -02
  .19000+01      .1073 -02
  .20000+01      .1044 -02
  .21000+01      .9972 -03
  .22000+01      .9654 -03
  .23000+01      .9393 -03
  .24000+01      .9199 -03
  .25000+01      .9077 -03
  .26000+01      .9044 -03
  .27000+01      .9042 -03
  .28000+01      .9146 -03
  .29000+01      .9312 -03
  .30000+01      .9555 -03
  .31000+01      .9774 -03
  .32000+01      .9961 -03
  .33000+01      .1059 -02
  .34000+01      .1099 -02
  .35000+01      .1141 -02
  .36000+01      .1194 -02
  .37000+01      .1224 -02
  .38000+01      .1262 -02
  .39000+01      .1297 -02
  .40000+01      .1324 -02
  .41000+01      .1334 -02
  .42000+01      .1339 -02
  .43000+01      .1371 -02
  .44000+01      .1372 -02
  .45000+01      .1364 -02
  .46000+01      .1352 -02
  .47000+01      .1331 -02
  .48000+01      .1314 -02
  .49000+01      .1272 -02
```



Sample Problem #4: Compatible voltage source loop.



$R_4 = 2\ \Omega$

$R_5 = 1\ \Omega$

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.

  set TSTART = TEND = 0.0

- desire error to fall below 10^{-5}

  $(EPS = 10^{-5})$

- the time increment is arbitrary

  (set H = 0.0)

- desire results, at each time point, to be outputed

  (NSTEP = 1)

- desire tree and $\underset{\wedge}{F}$ to be outputed

  (NCONT = 1)

- desire automatic internal scaling

  $(NSCALE \neq 1)$

- desire all branch currents and voltages to be 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 0.23 seconds.

DATA CARDS: PROBLEM #4

```
    4    6            0.0            0.0            0.0    1        0.00001    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
```

Comments on Sample Problem #4:

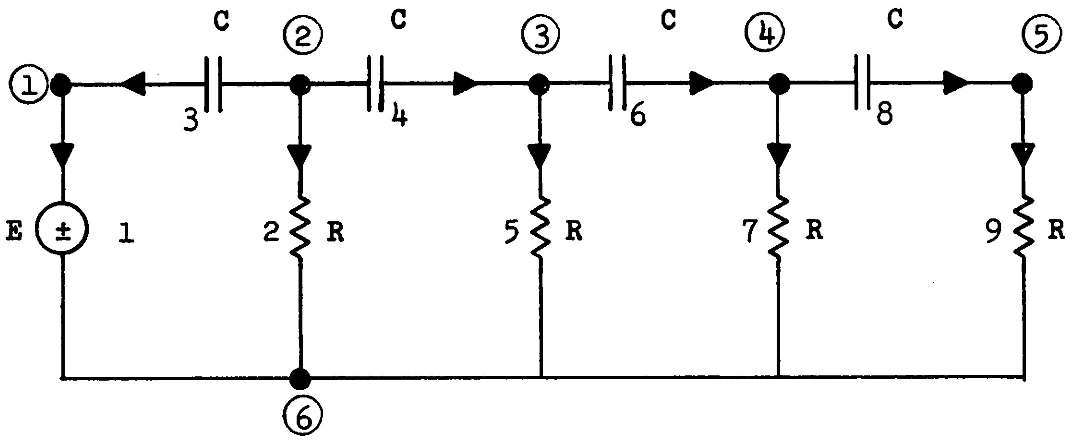
Note that when dependent and independent voltage source loops exist in a network, it is of paramount importance for them to be compatible, i.e., that Kirchhoff's voltage law be satisfied around that loop.

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

```
TIME =      0.
 BRANCH     1      CURRENT                 -.10000E+01
 BRANCH     2      CURRENT                 -.10000E+01
 BRANCH     3      CURRENT                 0.
 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:



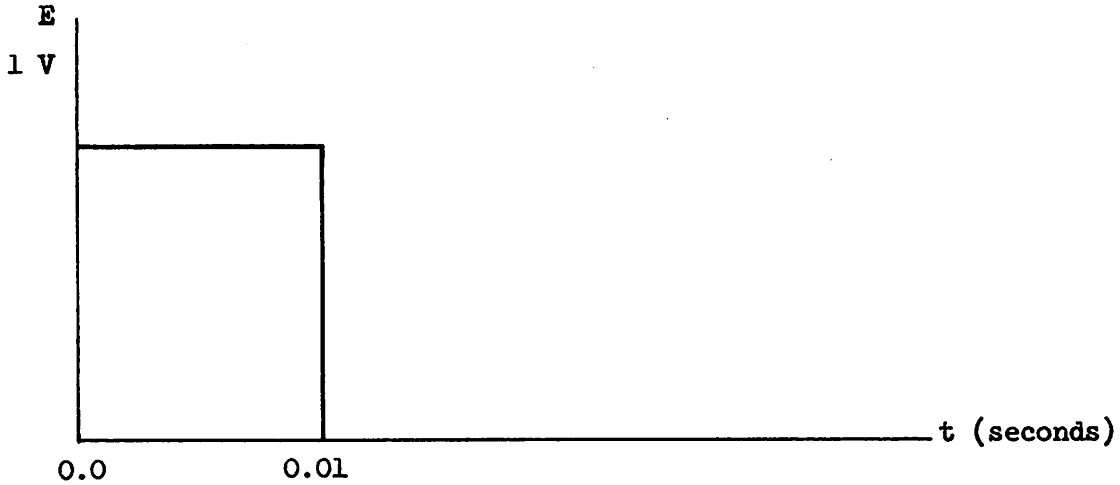
R = 10^3

C = 10^{-5}

NN = 6

NB = 9

The input E is specified as follows



Specifications:

- initial time = 0 (TSTART = 0.0)
- final time = 50 millisec (TEND = 0.05)
- time increment = 1 millisec (H = 0.001)
- desire 10 iterative steps to be taken at each time point

  (EPS = 0.0, NITT = 10)
- desire results, at each time point, to be outputed (NSTEP = 1)
- do not desire tree information and $\underset{\wedge}{F}$ to be outputed (NCONT = 0)
- desire internal automatic current scaling (NSCALE $\neq$ 1)
- zero initial conditions
- desire the following graphical outputs

$$\left.\begin{array}{ll}\text{Branch 1} & \text{voltage}\\ \text{Branch 5} & \text{voltage}\end{array}\right\}\ \text{(NGRAPH = 2)}$$

The central processor time, for this problem, was 8.26 seconds.

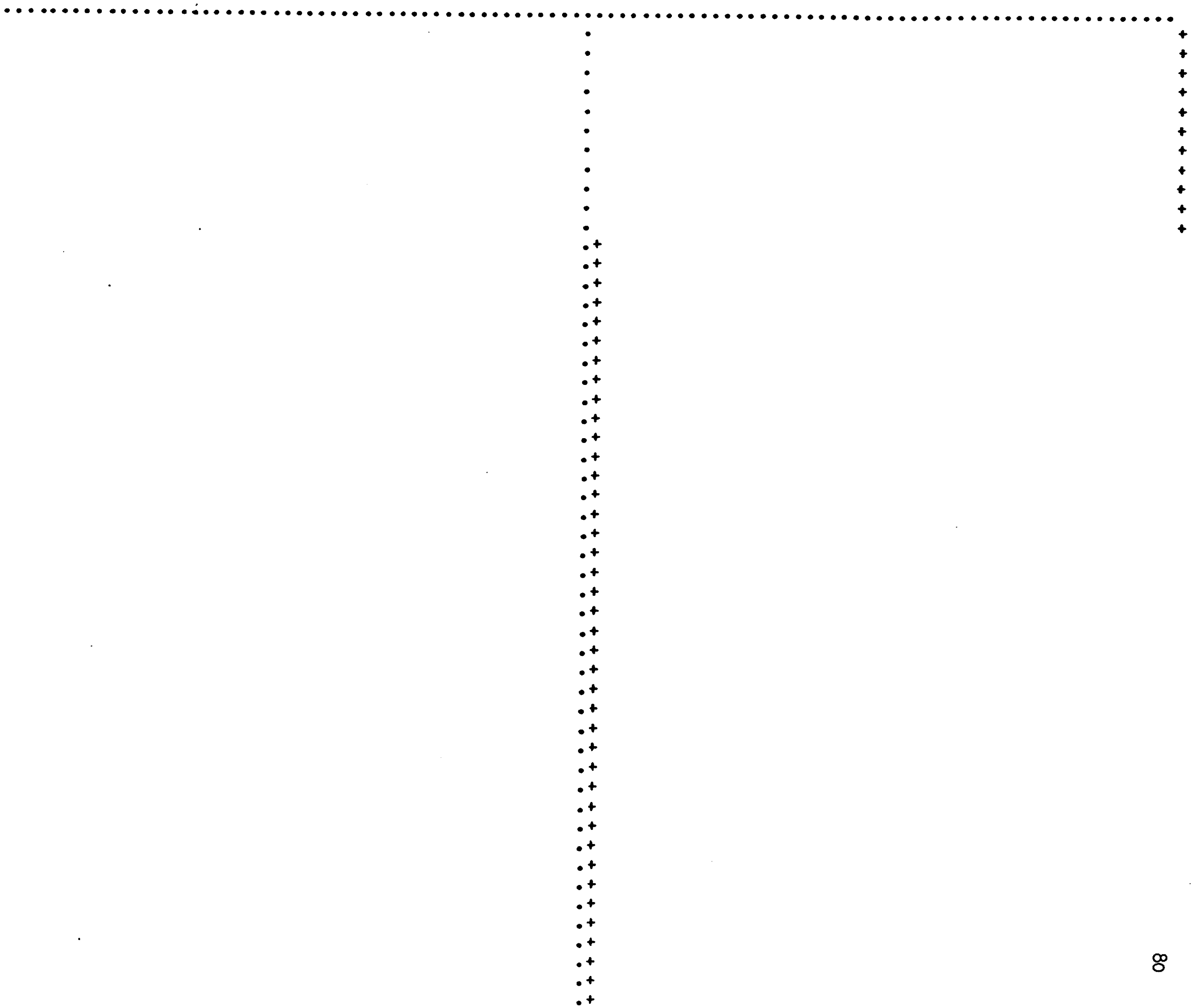
The required data cards, for this problem, may be seen on the following page.

DATA CARDS : PROBLEM #5

```
    6    9            0.0           0.05          0.001    1
    4                            10
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
```

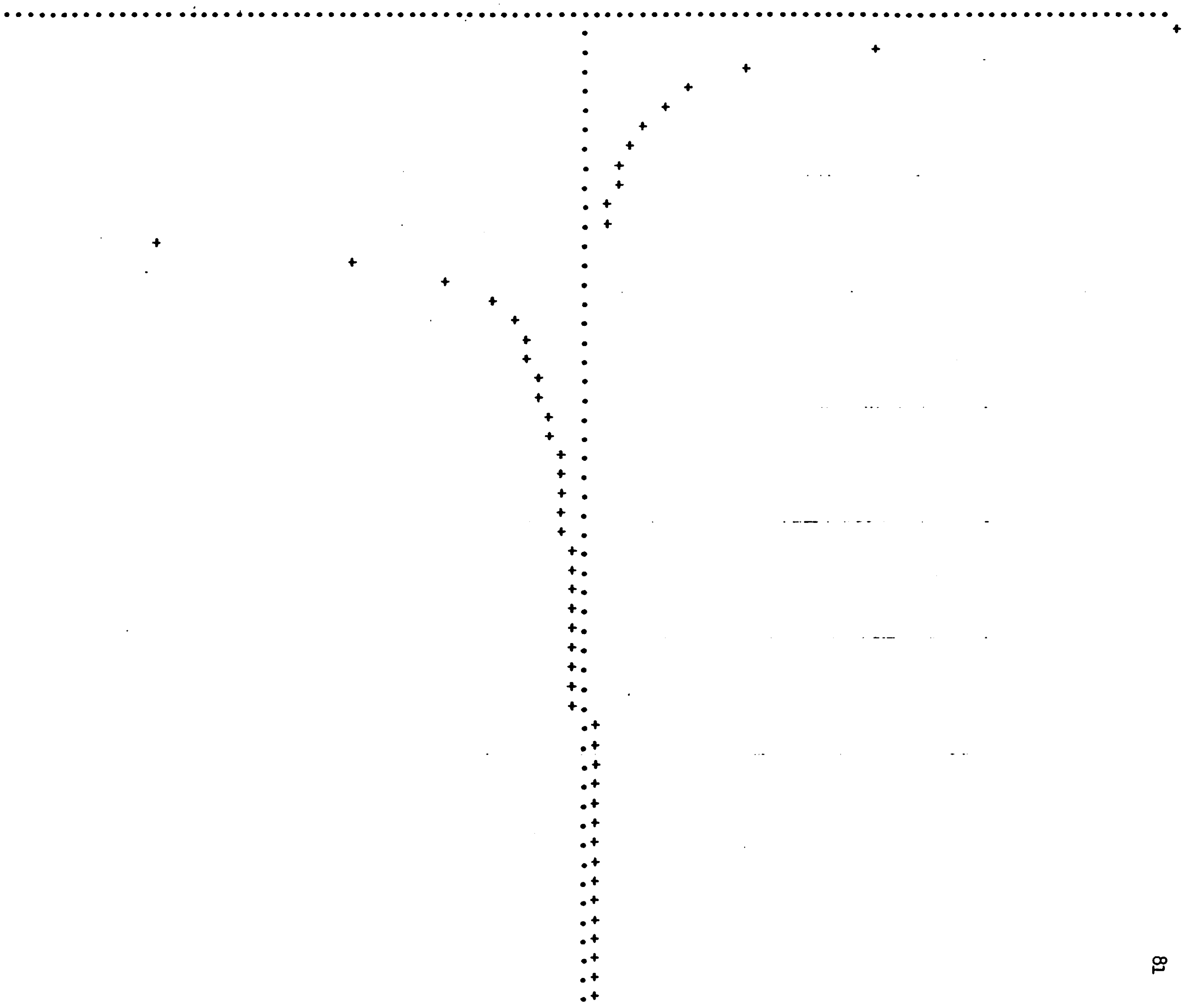
```
                                BRANCH   1      VOLTAGE

 TIME             VALUE
0.               .10000E+01
 .10000E-02      .10000E+01
 .20000E-02      .10000E+01
 .30000E-02      .10000E+01
 .40000E-02      .10000E+01
 .50000E-02      .10000E+01
 .60000E-02      .10000E+01
 .70000E-02      .10000E+01
 .80000E-02      .10000E+01
 .90000E-02      .10000E+01
 .10000E-01      .10000E+01
 .11000E-01    0.
 .12000E-01    0.
 .13000E-01    0.
 .14000E-01    0.
 .15000E-01    0.
 .16000E-01    0.
 .17000E-01    0.
 .18000E-01    0.
 .19000E-01    0.
 .20000E-01    0.
 .21000E-01    0.
 .22000E-01    0.
 .23000E-01    0.
 .24000E-01    0.
 .25000E-01    0.
 .26000E-01    0.
 .27000E-01    0.
 .28000E-01    0.
 .29000E-01    0.
 .30000E-01    0.
 .31000E-01    0.
 .32000E-01    0.
 .33000E-01    0.
 .34000E-01    0.
 .35000E-01    0.
 .36000E-01    0.
 .37000E-01    0.
 .38000E-01    0.
 .39000E-01    0.
 .40000E-01    0.
 .41000E-01    0.
 .42000E-01    0.
 .43000E-01    0.
 .44000E-01    0.
 .45000E-01    0.
 .46000E-01    0.
 .47000E-01    0.
 .48000E-01    0.
 .49000E-01    0.
 .50000E-01    0.
```



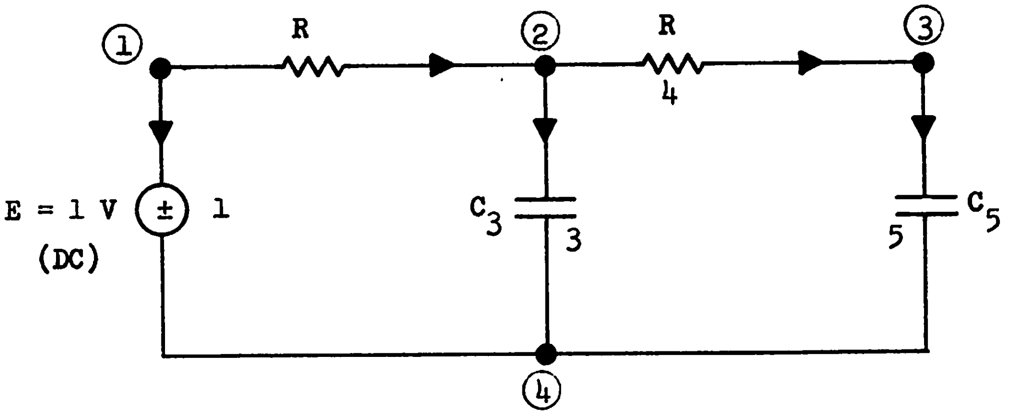
BRANCH 5 VOLTAGE

```
 TIME          VALUE
0.            .1000E+01
 .1000E-02    .4983E+00
 .2000E-02    .2778E+00
 .3000E-02    .1750E+00
 .4000E-02    .1222E+00
 .5000E-02    .9134E-01
 .6000E-02    .7054E-01
 .7000E-02    .5445E-01
 .8000E-02    .4133E-01
 .9000E-02    .3028E-01
 .1000E-01    .2071E-01
 .1100E-01   -.7368E+00
 .1200E-01   -.3830E+00
 .1300E-01   -.2277E+00
 .1400E-01   -.1555E+00
 .1500E-01   -.1185E+00
 .1600E-01   -.9672E-01
 .1700E-01   -.8204E-01
 .1800E-01   -.7067E-01
 .1900E-01   -.6126E-01
 .2000E-01   -.5322E-01
 .2100E-01   -.4616E-01
 .2200E-01   -.3990E-01
 .2300E-01   -.3433E-01
 .2400E-01   -.2935E-01
 .2500E-01   -.2492E-01
 .2600E-01   -.2097E-01
 .2700E-01   -.1745E-01
 .2800E-01   -.1431E-01
 .2900E-01   -.1153E-01
 .3000E-01   -.9055E-02
 .3100E-01   -.6864E-02
 .3200E-01   -.4927E-02
 .3300E-01   -.3217E-02
 .3400E-01   -.1711E-02
 .3500E-01   -.3889E-03
 .3600E-01    .7689E-03
 .3700E-01    .1779E-02
 .3800E-01    .2657E-02
 .3900E-01    .3417E-02
 .4000E-01    .4071E-02
 .4100E-01    .4630E-02
 .4200E-01    .5105E-02
 .4300E-01    .5504E-02
 .4400E-01    .5836E-02
 .4500E-01    .6109E-02
 .4600E-01    .6329E-02
 .4700E-01    .6501E-02
 .4800E-01    .6631E-02
 .4900E-01    .6725E-02
 .5000E-01    .6786E-02
```



Sample Problem #6:

Large time constant spread problem--quick response:



$R = 10^3$
$C_3 = 10^{-8}$
$C_5 = 0.1$

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 microseconds (TEND = 5×10^{-5})
- time increment = 1 microsecond (H = 10^{-6})
- desire error to fall below 10^{-14}
  (EPS = 10^{-14})
- desire results, at each time point, to be outputed (NSTEP = 1)
- do not desire tree information and $\underset{\wedge}{F}$ to be outputed (NCONT ≠ 1)
- desire internal, automatic current scaling (NSCALE ≠ 1)
- zero initial conditions

- desire the following graphical outputs

$$\left.\begin{array}{ll}\text{Branch 3} & \text{voltage}\\ \text{Branch 4} & \text{current}\\ \text{Branch 2} & \text{current}\\ \text{Branch 5} & \text{voltage}\end{array}\right\} \qquad (\text{NGRAPH} = 4)$$

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

The central processor time, for this problem, was 4.11 seconds.

DATA CARDS : PROBLEM #6

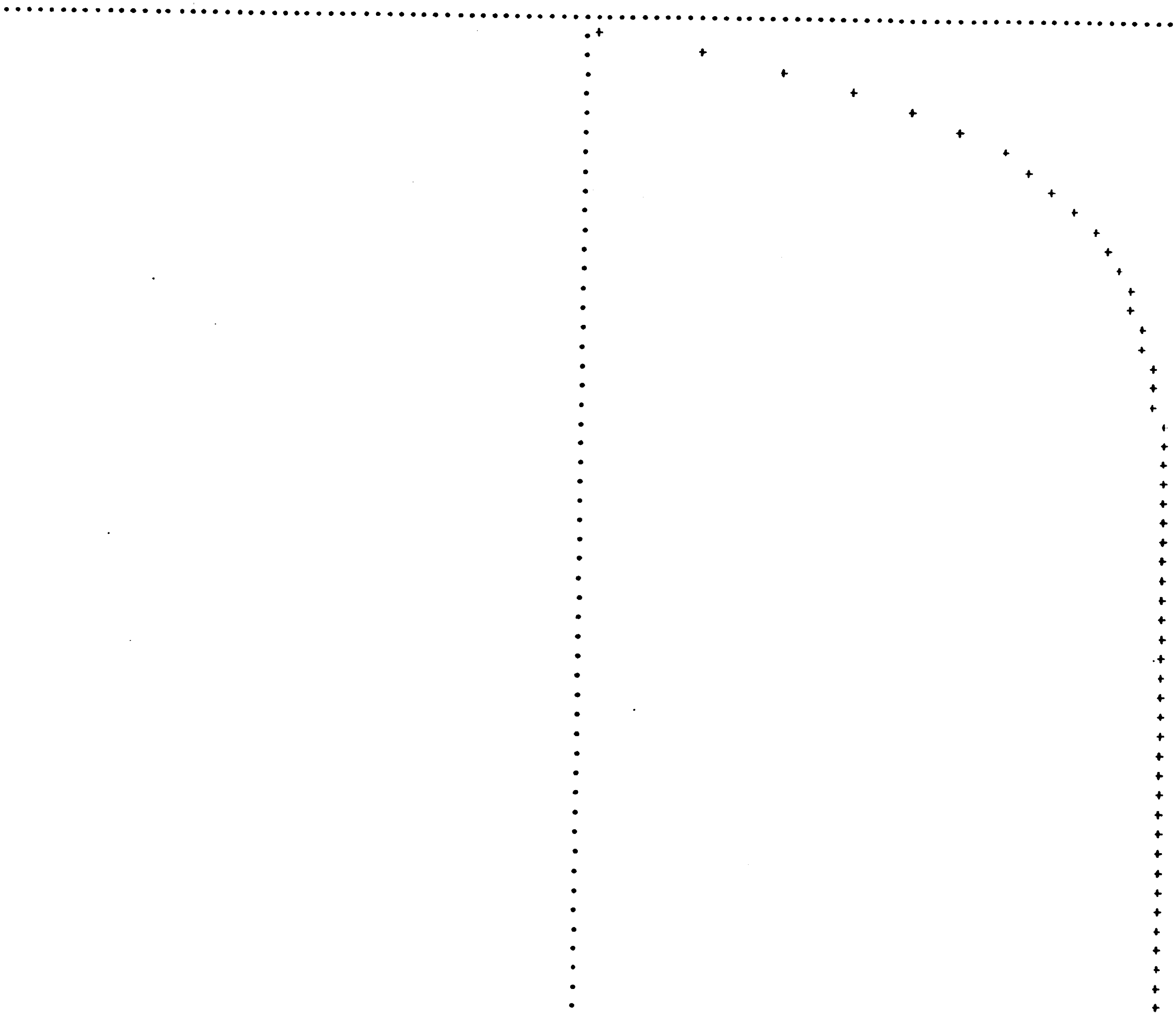
```
1234567890123456789012345678901234567890123456789012345678901234567890123456789 80
    4    5            0.0        9.5E-04        1.0E-06    1        1.0E-14
    4
E  1C      1  4                                1.0
R   2      1  2              1000.0
C   3      2  4             1.0E-08
R   4      2  3              1000.0
C   5      3  4                 0.1
    3    0
    4    1
    2    1
    5    0
```

Comments on Sample Problem #6:

The most important thing to notice in the computer output of this problem is that the voltage across C_5 is, essentially, zero; i.e., the large capacitance is acting as a short circuit over the time interval of concern.

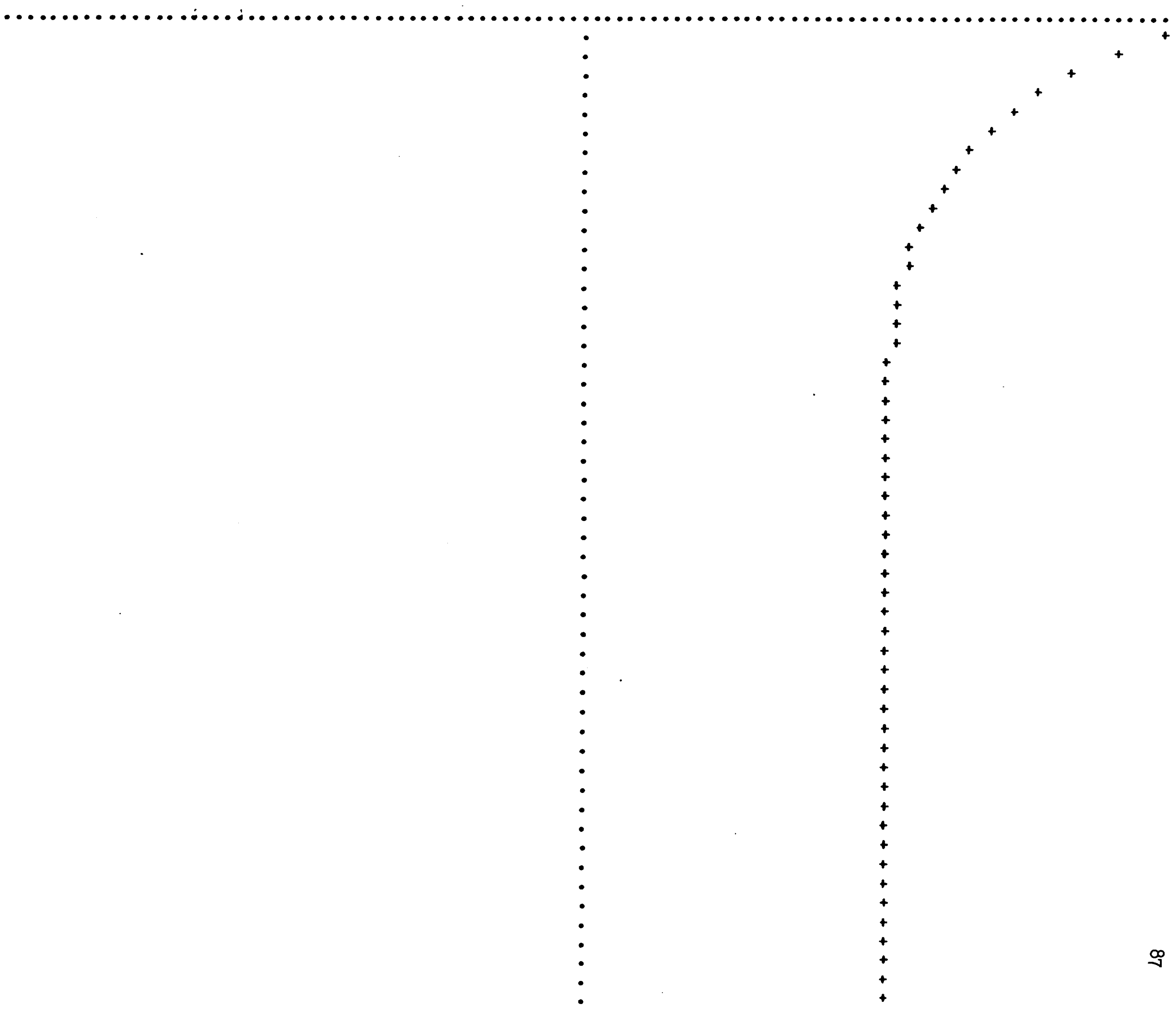
The quick response steady state voltage across C_3 is 0.5 volts, and, as will be noted in problem #7, this value must be used as the initial condition on C_3 when the slow time response is desired.

```
                    BRANCH    3       VOLTAGE
  TIME           VALUE
C.              0.
 .1000E-05      .9991E-01
 .2000E-05      .1653E+00
 .3000E-05      .2251E+00
 .4000E-05      .2759E+00
 .5000E-05      .3157E+00
 .6000E-05      .3500E+00
 .7000E-05      .3773E+00
 .8000E-05      .3996E+00
 .9000E-05      .4179E+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 error to fall below 10^{-14} or 20 iteration steps, per time point. (NITT = 20, EPS = 10^{-14})
- desire results, at each time point, to be outputed (NSTEP = 1)
- do not desire tree information and $\underset{\wedge}{F}$ to be outputed (NCONT = 0)
- desire internal, automatic current scaling (NSCALE $\neq$ 1)
- initial voltages are

$$V_{C3}(0) = 0.5 \text{ V}$$

$$V_{C5}(0) = 0 \text{ V}$$

- desire the following graphical outputs

$$\left.\begin{array}{ll}\text{Branch 3} & \text{voltage}\\ \text{Branch 4} & \text{current}\\ \text{Branch 2} & \text{current}\\ \text{Branch 5} & \text{voltage}\end{array}\right\} \qquad \text{(NGRAPH = 4)}$$

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

The central processor time, for this problem, was 3.80 seconds.

DATA CARDS : PROBLEM # 7

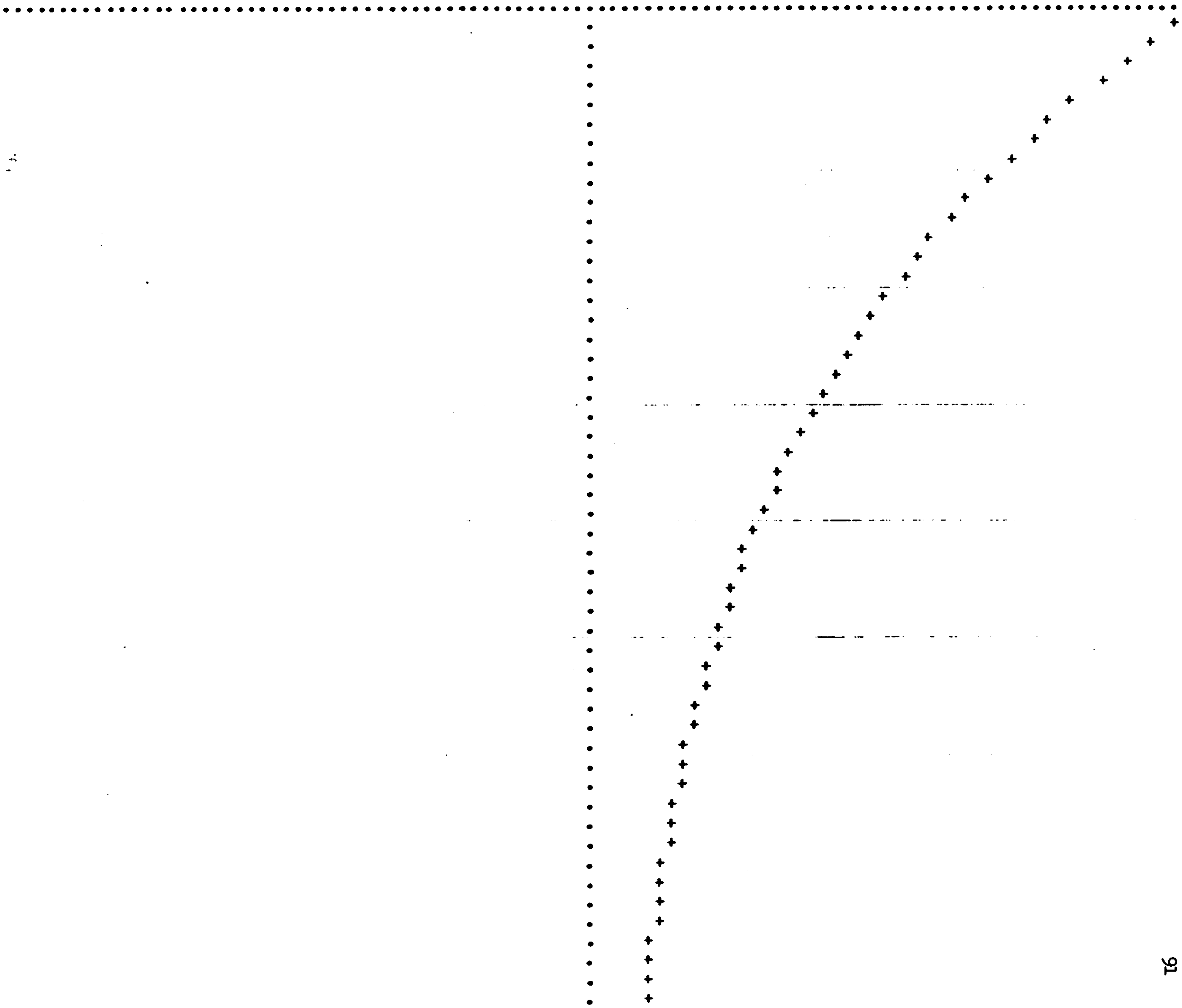
```
    4    5           10.0          500.0           10.0    1        1.0E-14
    4                            20
E  1C      1  4                                1.0
R  2       1  2              1000.0
C  3       2  4             1.0E-08            0.5
R  4       2  3              1000.0
C  5       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.

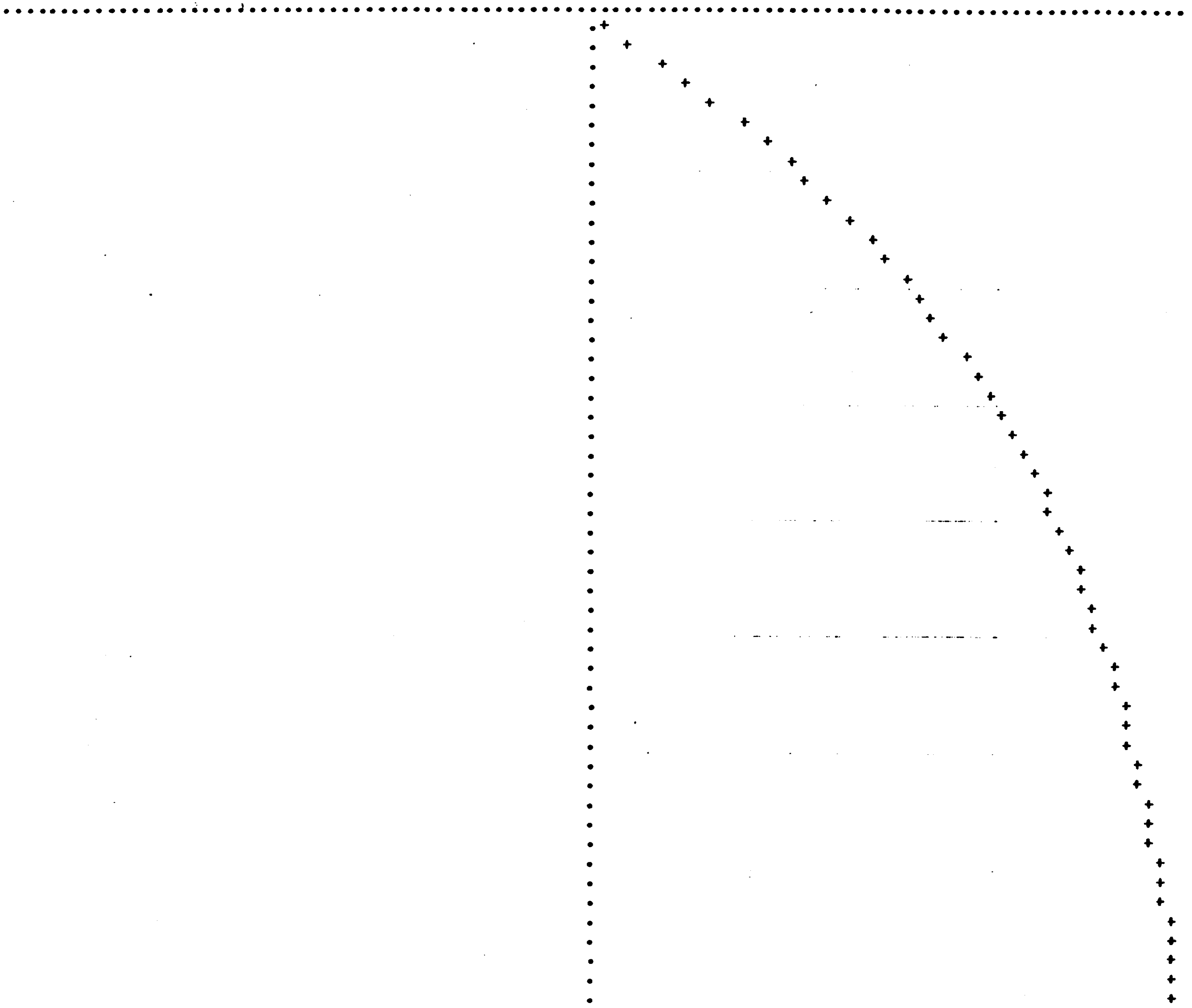
```
                                BRANCH    2       CURRENT

 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
```



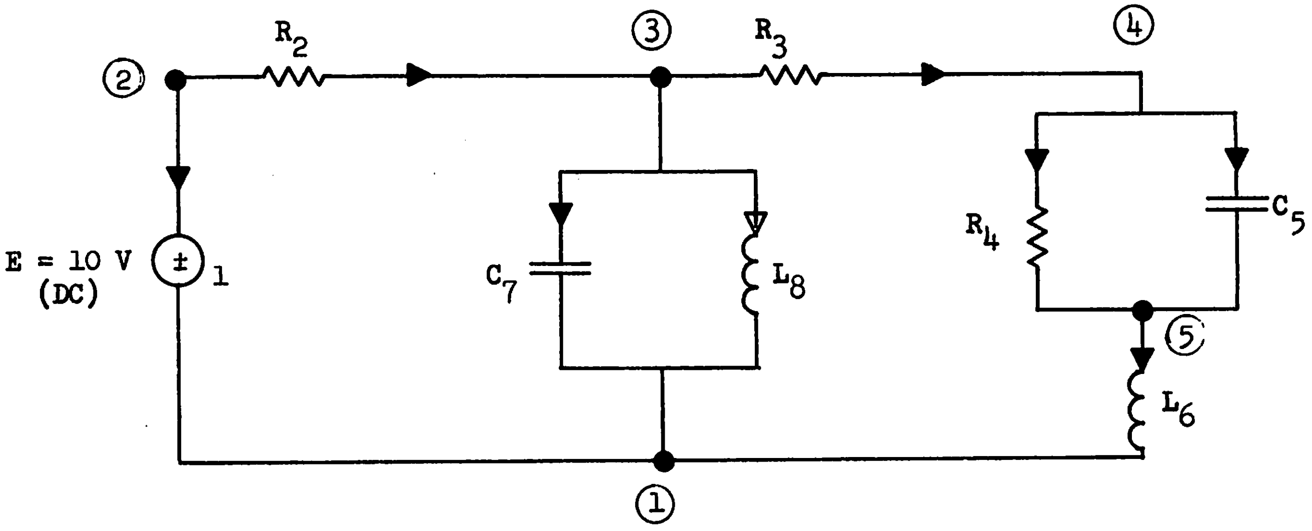
BRANCH 5 VOLTAGE

```
 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-valued 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 $\underset{\wedge}{F}$ to be outputed (NCONT = 1)
- desire error at each time point to fall below 10^{-6} (EPS = 10^{-6})
- desire all voltages and currents outputed at every fiftieth time point (NSTEP = 50)

- desire internal automatic current scaling (NSCALE $\neq$ 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 3.48 seconds.

DATA CARDS : PROBLEM #8

```
    5    8            0.0            5.1            0.1   50      0.000001
             16
E  1C      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    1
    7    1
    8    1
```

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 valued 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 satisfied, an arithmetic error will result during the execution of the program.

```
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                    .10058E+02
 BRANCH      3        VOLTAGE                   -.55976E-03
 BRANCH      4        VOLTAGE                   -.56889E-01
 BRANCH      5        VOLTAGE                   -.56889E-01
 BRANCH      6        VOLTAGE                   -.10373E-02
 BRANCH      7        VOLTAGE                   -.58486E-01
 BRANCH      8        VOLTAGE                   -.58486E-01
 BRANCH      1        CURRENT                   -.10058E+02
 BRANCH      2        CURRENT                    .10058E+02
 BRANCH      3        CURRENT                   -.28288E-01
 BRANCH      4        CURRENT                   -.28275E-01
 BRANCH      5        CURRENT                   -.13688E-04
 BRANCH      6        CURRENT                   -.28288E-01
 BRANCH      7        CURRENT                   -.18666E+00
 BRANCH      8        CURRENT                    .10273E+02
```

APPENDIX D:

CANDO FORTRAN IV LISTING

(CDC 6400)

```
      PROGRAM CANDO(INPUT,OUTPUT)
      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),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 /BLOCK17/ ITN,S(200)
      COMMON /BLOCK18/ VAROUT(200)
      COMMON /BLOCK19/ CNTSOR(200)
      COMMON /BLOCK20/ NCONT,KLOOP,KPP
      COMMON /BLOCK21/ NST(5)
      COMMON /BLOCK23/ TEMPAR(200),DUMMY(200)
      COMMON /BLOCK24/ SCAL,NSCALE
      COMMON /BLOCK25/ X(200),Y(200)
      DIMENSION HH(20,20)
      NDATA=0
      NLNEAR=0
    1 NIT=0
      ITN=1
      NPRINT=0
      NIC=0
      IF (NDATA.EQ.0) GO TO 2
      CALL SECOND (T)
      TO=T-TU
      PRINT 16, 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
      ITN=1
      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
      IF (NIC.EQ.1) CALL KONST
      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 CALCAL
      IF (NITT.EQ.0) GO TO 12
      IF (NIT-NITT) 12,14,14
   12 CALL ERROR
      IF (FUNCT-EPS) 14,14,13
   13 CALL FNDGRAD
      CALL FLPOW (HH,NB,ALPHA)
      ITN=ITN+1
      NIT=NIT+1
      CALL FNDALPH
      CALL CNGVARS
      GO TO 11
   14 IDEL=IDEL+1
      IF (IDEL.NE.NSTEP) GO TO 15
      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
   15 IF (NIC-NITER) 3,1,1
C
   16 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 CONTYPE,SORTYPE
      INTEGER SORTEM,CONTTEM
      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 18, NN,NB,TSTART,TEND,H,NSTEP,EPS,NCONT
      IF (NN.EQ.0) STOP
      READ 19, NGRAPH,NALLOUT,JOUT,SCALE,NITT,NSCALE
      PRINT 20, EPS,TSTART,TEND,H
      PRINT 21, NN,NB,SCALE,NITT,NSCALE
      PRINT 22, 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 23, 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 24, (SNSOID(KSOR,J),J=1,3)
      GO TO 9
6     KSOR=KSOR+1
      NNJ(KSOR)=0
      READ 25, NNI(KSOR)
      NNN=NNI(KSOR)
      READ 26, (SORVAL(KSOR,J),J=1,NNN)
      READ 26, (TIMEPT(KSOR,J),J=1,NNN)
      GO TO 9
7     KCURR=KCURR-1
      NNJ(KCURR)=0
      READ 24, (SNSOID(KCURR,J),J=1,3)
      GO TO 9
8     KCURR=KCURR-1
      NNJ(KCURR)=0
      READ 25, NNI(KCURR)
      NNN=NNI(KCURR)
      READ 26, (SORVAL(KCURR,J),J=1,NNN)
      READ 26, (TIMEPT(KCURR,J),J=1,NNN)
9     CONTINUE
      IF (NGRAPH.EQ.0) GO TO 10
      READ 27, ((IGRAPH(I),JGRAPH(I)),I=1,NGRAPH)
10    IF (NALLOUT.GE.1.OR.NGRAPH.GE.1) GO TO 11
      READ 27, ((IOUT(I),ITEST(I)),I=1,JOUT)
11    PRINT 28
      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 (I5,I5,E15.3,E15.3,E15.3,I5,E15.3,I5)
19    FORMAT (3I5,E15.5,I5,I5)
20    FORMAT (1H1,/////,40X,* ERROR CRITERION = *,E15.5,///,40X,* STARTI
     1NG TIME = *,E15.5,*   END TIME = *,E15.5,///,40X,*STEP SIZE = *,
     2 E15.5)
21    FORMAT (1H0,///,30X,*NUMBER OF NODES = *,I5,*    NUMBER OF BRANCHE
     1S = *,I5,///,30X,*SCALE FACTOR = *,E15.5,///,30X,* DESIRED NUMBER
     2OF ITERATIONS AT EACH TIME POINT = *,I5,///,30X,*NSCALE = *,I5)
22    FORMAT (1H0,///,30X,* NUMBER OF TIME ITERATIONS PER OUTPUT = *,I5)
23    FORMAT (A1,I3,A1,A1,I3,1X,I2,1X,I2,1X,I2,2X,E15.4,E15.4)
24    FORMAT ( 3E10.3)
25    FORMAT ( I5 )
26    FORMAT ( 8E10.3 )
27    FORMAT (2I5)
28    FORMAT (1H1,55X,*THIS IS THE GIVEN NETWORK*)
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 TEMPO1,TEMPO2,TEMPO3,TEMPO4,TEMPO6,TEMPO7,TEMPO9
      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)
      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)
      TEMP10=CONST(I)
      IF (NIC.EQ.0) GO TO 12
      TEMP11=V(I,2)                                                     C 110
      TEMP12=C(I,2)                                                     C 111
      TEMP13=V(I,1)                                                     C 112
      TEMP14=C(I,1)                                                     C 113
   12 ITBRAN(I)=ITBRAN(J)                                               C 114
      LEAVT(I)=LEAVT(J)                                                 C 115
      LENTT(I)=LENTT(J)                                                 C 116
      TYPE(I)=TYPE(J)                                                   C 117
      VALUE(I)=VALUE(J)                                                 C 118
      CONTYPE(I)=CONTYPE(J)                                             C 119
      SORTYPE(I)=SORTYPE(J)                                             C 120
      COND(I)=COND(J)                                                   C 121
      KONBRAN(I)=KONBRAN(J)                                             C 122
      CONST(I)=CONST(J)                                                 C 123
      IF (NIC.EQ.0) GO TO 13                                            C 124
      V(I,2)=V(J,2)                                                     C 125
      C(I,2)=C(J,2)                                                     C 126
      V(I,1)=V(J,1)                                                     C 127
      C(I,1)=C(J,1)                                                     C 128
   13 ITBRAN(J)=TEMPO1                                                  C 129
      LEAVT(J)=TEMPO2                                                   C 130
      LENTT(J)=TEMPO3                                                   C 131
      TYPE(J)=TEMPO4                                                    C 132
      VALUE(J)=TEMPO5                                                   C 133
      CONTYPE(J)=TEMPO6                                                 C 134
      SORTYPE(J)=TEMPO7                                                 C 135
      COND(J)=TEMPO8                                                    C 136
      KONBRAN(J)=TEMPO9                                                 C 137
      CONST(J)=TEMP10                                                   C 138
      IF (NIC.EQ.0) GO TO 14                                            C 139
      V(J,2)=TEMP11                                                     C 140
      C(J,2)=TEMP12                                                     C 141
      V(J,1)=TEMP13                                                     C 142
      C(J,1)=TEMP14                                                     C 143
   14 CONTINUE                                                          C 144
   15 IF (2.GT.KK) GO TO 32                                             C 144A
      DO 25 I=2,KK                                                      C 145
   16 NE=LENTT(I)                                                       C 146
      NL=LEAVT(I)                                                       C 147
      NP(1)=I                                                           C 148
      NNP(1)=1                                                          C 149
      M=I-1                                                             C 150
      MT=0                                                              C 151
      KT=1                                                              C 152
   17 DO 19 JJJ=1,M                                                     C 153
      J=M+1-JJJ                                                         C 154
      IF (MT.EQ.J) GO TO 18                                             C 155
      IF (LEAVT(J).EQ.NE) GO TO 23                                      C 156
      IF (LENTT(J).EQ.NE) GO TO 24                                      C 157
   18 CONTINUE                                                          C 158
   19 CONTINUE                                                          C 159
   20 IF (KT.EQ.1) GO TO 25                                             C 160
      KA=NNP(KT)                                                        C 161
      KB=NP(KT)                                                         C 162
      M=KB-1                                                            C 163
      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 LAF=NOEL(1)
      IF (I.LE.LAF) 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-NOFL(7)
      ITBRAN(NZ)=TEMPO1
      LEAVT(NZ)=TEMPO2
      LENTT(NZ)=TEMPO3
      TYPE(NZ)=TEMPO4
      VALUE(NZ)=TEMPO5
      CONTYPE(NZ)=TEMPO6
      SORTYPE(NZ)=TEMPO7
      COND(NZ)=TEMPO8
      KONBRAN(NZ)=TEMPO9
      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   *,E12.5,*  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 TYTEMP
      INTEGER TYPE
      INTEGER TEMP
      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,* F PORTION OF FUNDAMENTAL CUTSET MATRIX *,6X,*PAG
     1F*,I5,//)
   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 TYPE
      INTEGER TEMP
      INTEGER CONTYPE,SORTYPE
      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)
      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)
        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
        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
      IF (NIC.EQ.0) GO TO 20
      RETURN
20    DO 21 I=1,NB
      IF (TYPE(I).EQ.1HL) CONST(I)=0.0
      IF (TYPE(I).EQ.1HC) CONST(I)=0.0
21    CONTINUE
      RETURN
      END
      SUBROUTINE CALCAL
      INTEGER TYPE
      INTEGER CNTSOR
      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 TYPE
      INTEGER TEMP
      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
      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
      TIME=TSTART+NIC*H
      M=NOEL(1)+NOEL(7)
      K=0
      DO 14 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 3
      GO TO 4
3     BOOL=.FALSE.
      K=5-ISTEP
4     IF (SORTYPE(J).EQ.1HC) GO TO 14
      K=K+1
      IF (SORTYPE(J).EQ.1HE.OR.SORTYPE(J).EQ.1HS) GO TO 7
      IF (SORTYPE(J).EQ.1HP) GO TO 11
15    IF (TIME.LE.SECTIME(K)) GO TO 5
      IF (NNI(K).LE.NP(K)) GO TO 12
      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)
      GO TO 15
5     IF (I.GE.NOEL(1)+1) GO TO 6
      V(J,2)=OLDVAL(K)+(SECVAL(K)-OLDVAL(K))*(TIME-OLDTIME(K))/(SECTIME(
     1K)-OLDTIME(K))
      GO TO 14
6     C(J,2)=OLDVAL(K)+(SECVAL(K)-OLDVAL(K))*(TIME-OLDTIME(K))/(SECTIME(
     1K)-OLDTIME(K))
      GO TO 14
7     W=SNSOID(K,2)
      T=SNSOID(K,3)
      IF (SORTYPE(J).EQ.1HS) GO TO 9
      IF (J.GT.I) GO TO 8
      V(J,2)=SNSOID(K,1)*EXP(W*TIME-T)
      GO TO 14
8     C(J,2)=SNSOID(K,1)*EXP(W*TIME-T)
      GO TO 14
9     IF (J.GT.I) GO TO 10
      V(J,2)=SNSOID(K,1)*SIN(W*TIME-T)
      GO TO 14
10    C(J,2)=SNSOID(K,1)*SIN(W*TIME-T)
      GO TO 14
11    CALL PERIOD (K,J)
      GO TO 14
12    L=NNI(K)
      IF (J.GT.I) GO TO 13
      V(J,2)=SORVAL(K,L)
      GO TO 14
13    C(J,2)=SORVAL(K,L)
14    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)
6     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)
      GO TO 6
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 ERROR
      INTEGER TYPE
      INTEGER TEMP
      INTEGER SORTYPE,CONTYPE
      INTEGER CNTSOR
      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 /BLOCK19/ CNTSOR(200)
      M=NOEL(1)+1
      N=NB-NOEL(7)
      DO 13 I=M,N
      IF (TYPE(I).EQ.1HV) GO TO 3
      IF (TYPE(I).EQ.1HC) GO TO 5
      IF (TYPE(I).EQ.1HR) GO TO 7
      IF (TYPE(I).EQ.1HL) GO TO 8
      IF (TYPE(I).EQ.1HI) GO TO 1
      E(I)=0.0
      GO TO 13
 1    K=CNTSOR(I)
      IF (CONTYPE(I).NE.1HV) GO TO 2
      E(I)=-C(I,2)+V(K,2)*CONST(I)
      GO TO 13
 2    E(I)=-C(I,2)+C(K,2)*CONST(I)
      GO TO 13
 3    K=CNTSOR(I)
      IF (CONTYPE(I).NE.1HV) GO TO 4
      E(I)=-V(I,2)+V(K,2)*CONST(I)
      GO TO 13
 4    E(I)=-V(I,2)+C(K,2)*CONST(I)
      GO TO 13
 5    IF (NIC.EQ.0) GO TO 6
      IF (I.GE.NN) GO TO 10
      E(I)=-(V(I,2)-V(I,1))+(C(I,2)+C(I,1))*CONST(I)
      GO TO 13
 6    E(I)=0.0
      GO TO 13
 7    IF (I.GE.NN) GO TO 11
      E(I)=-V(I,2)+C(I,2)*CONST(I)
      GO TO 13
 8    IF (NIC.EQ.0) GO TO 9
      IF (I.GE.NN) GO TO 12
      E(I)=-(V(I,2)+V(I,1))+(C(I,2)-C(I,1))*CONST(I)
      GO TO 13
 9    E(I)=0.0
      GO TO 13
 10   E(I)=-(C(I,2)+C(I,1))+(V(I,2)-V(I,1))*CONST(I)
      GO TO 13
 11   E(I)=-C(I,2)+V(I,2)*CONST(I)
      GO TO 13
 12   E(I)=-(C(I,2)-C(I,1))+(V(I,2)+V(I,1))*CONST(I)
 13   CONTINUE
      FUNCT=0.0
      DO 14 I=M,N
      FUNCT=FUNCT+E(I)**2
 14   CONTINUE
      FUNCT=FUNCT/2
      RETURN
      END
      SUBROUTINE FNDGRAD
      INTEGER CNTSOR
      INTEGER TYPE
      INTEGER TEMP
      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 /BLOCK19/ CNTSOR(200)
      COMMON /BLOCK23/ TEMPAR(200),DUMMY(200)
      M=NOEL(1)+1
      MM=NB-NOEL(7)
      N=NN-1
      DO 5 I=1,N
      IF (TYPE(I).EQ.1HE) GO TO 1
      IF (TYPE(I).EQ.1HI) GO TO 3
      IF (TYPE(I).EQ.1HV) GO TO 4
      TEMPAR(I)=-E(I)
      DUMMY(I)=CONST(I)*E(I)
      GO TO 5
  1   K=CNTSOR(I)
      IF (K.EQ.0) GO TO 2
      TEMPAR(I)=0.0
      DUMMY(I)=CONST(K)*E(K)
      GO TO 5
  2   TEMPAR(I)=0.0
      DUMMY(I)=0.0
      GO TO 5
  3   TEMPAR(I)=0.0
      DUMMY(I)=-E(I)
      GO TO 5
  4   TEMPAR(I)=-E(I)
      DUMMY(I)=0.0
  5   CONTINUE
      DO 10 I=NN,NB
      IF (TYPE(I).EQ.1HJ) GO TO 6
      IF (TYPE(I).EQ.1HV) GO TO 8
      IF (TYPE(I).EQ.1HI) GO TO 9
      TEMPAR(I)=CONST(I)*E(I)
      DUMMY(I)=-E(I)
      GO TO 10
  6   K=CNTSOR(I)
      IF (K.EQ.0) GO TO 7
      TEMPAR(I)=CONST(K)*E(K)
      DUMMY(I)=0.0
      GO TO 10
  7   TEMPAR(I)=0.0
      DUMMY(I)=0.0
      GO TO 10
  8   TEMPAR(I)=-E(I)
      DUMMY(I)=0.0
      GO TO 10
  9   TEMPAR(I)=0.0
      DUMMY(I)=-E(I)
 10   CONTINUE
      IF (NIC.NE.0) GO TO 13
      DO 12 I=M,MM
      IF (TYPE(I).EQ.1HC.OR.TYPE(I).EQ.1HL) GO TO 11
      GO TO 12
11    TEMPAR(I)=0.0
      DUMMY(I)=0.0
12    CONTINUE
13    IF (M.GT.N) GO TO 18
      DO 14 I=M,N
      GRAD(I)=TEMPAR(I)
      DO 14 J=NN,NB
14    GRAD(I)=GRAD(I)+F(I,J-N)*TEMPAR(J)
18    IF (NN.GT.MM) GO TO 19
      DO 15 I=NN,MM
      GRAD(I)=DUMMY(I)
      DO 15 J=1,N
15    GRAD(I)=GRAD(I)-F(J,I-N)*DUMMY(J)
19    IF (NIC.EQ.0) GO TO 16
      RETURN
16    DO 17 I=1,NB
      IF (TYPE(I).EQ.1HC) GRAD(I)=0.0
      IF (TYPE(I).EQ.1HL) GRAD(I)=0.0
17    CONTINUE
      RETURN
      END
      SUBROUTINE FNDALPH
C     THIS SUBROUTINE FINDS THE DISTANCE WE SHOULD GO ALOND THE GRADIENT
      INTEGER TYPE
      INTEGER TEMP
      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 /BLOCK12/ CONTTEM(200),SORTEM(200),CONDTEM(200),KONTEM(200)
      COMMON /BLOCK15/ NIT
      COMMON /BLOCK17/ ITN,S(200)
      FZERO=FUNCT
      M=NOEL(1)+1
      N=NB-NOEL(7)
      MM=NN-1
      PROD=0.0
      IF (M.GT.MM) GO TO 5
      DO 1 I=M,MM
      PROD=PROD+GRAD(I)*S(I)
      CONTTEM(I)=V(I,2)
      V(I,2)=V(I,2)-S(I)
    1 CONTINUE
      IF (NN.GT.N) GO TO 6
    5 DO 2 I=NN,N
      PROD=PROD+GRAD(I)*S(I)
      CONTTEM(I)=C(I,2)
      C(I,2)=C(I,2)-S(I)
    2 CONTINUE
    6 CALL CALCAL
      CALL ERROR
      FGRAD=FUNCT
      IF (M.GT.MM) GO TO 7
      DO 3 I=M,MM
      V(I,2)=CONTTEM(I)
    3 CONTINUE
    7 IF (NN.GT.N) GO TO 8
      DO 4 I=NN,N
      C(I,2)=CONTTEM(I)
    4 CONTINUE
    8 ALPHA=PROD/(FGRAD+PROD-FZERO)
      ALPHA=ALPHA/2.0
      RETURN
      END
      SUBROUTINE CNGVARS
C     THIS SUBROUTINE CHANGES THE CURRENTS ALONG THE GRADIENT BY THE
C     CONSTANT ALPHA
      INTEGER TYPE
      INTEGER TEMP
      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 /BLOCK17/ ITN,S(200)
      M=NOEL(1)+1
      N=NB-NOEL(7)
      IEND=NN-1
      IF (M.GT.IEND) GO TO 3
      DO 1 I=M,IEND
      V(I,2)=V(I,2)-ALPHA*S(I)
1     CONTINUE
      IF (NN.GT.N) RETURN
3     DO 2 I=NN,N
      C(I,2)=C(I,2)-ALPHA*S(I)
2     CONTINUE
      RETURN
      END
      SUBROUTINE FLPOW (HH,NB,ALPHA)
      INTEGER TEMP
      INTEGER TYPE
      COMMON /BLOCK3/ TYPE(200),VALUE(200),F(49,151),NOEL(7)
      COMMON /BLOCK6/ TEMP(7),E(200),GRAD(200),CONST(200)
      COMMON /BLOCK17/ ITN,S(200)
      COMMON /BLOCK25/ GRADB(200),DUMMY(200)
      DIMENSION HH(NB,NB)
      IF (ITN.GT.1) GO TO 3
      M=NOEL(1)+1
      N=NB-NOEL(7)
      DO 2 I=M,N
      DO 1 J=M,N
 1    HH(I,J)=0.0
      HH(I,I)=1.0
      GRADB(I)=GRAD(I)
 2    S(I)=GRAD(I)
      RETURN
 3    GHG=0.
      SG=0.
      DO 4 I=M,N
      S(I)=ALPHA*S(I)
      GRADB(I)=GRAD(I)-GRADB(I)
 4    SG=SG+S(I)*GRADB(I)
      DO 5 I=M,N
      DUMMY(I)=0.
      DO 5 J=M,N
      GHG=GHG+GRADB(I)*GRADB(J)*HH(I,J)
 5    DUMMY(I)=DUMMY(I)+HH(I,J)*GRADB(J)
      DO 7 I=M,N
      GRADB(I)=0.0
      DO 6 J=I,N
      HH(I,J)=HH(I,J)-(S(I)*S(J)/SG)-(DUMMY(I)*DUMMY(J)/GHG)
 6    HH(J,I)=HH(I,J)
      DO 7 J=M,N
 7    GRADB(I)=GRADB(I)+HH(I,J)*GRAD(J)
      DO 8 I=M,N
      S(I)=GRADB(I)
 8    GRADB(I)=GRAD(I)
      RETURN
      END
      SUBROUTINE READOUT
      INTEGER TYPE
      INTEGER TEMP
      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 ANP
      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 10
      A=7HVOLTAGE
      IF (JGRAPH(I).EQ.1) A=7HCURRENT
      PRINT 11, 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
```

APPENDIX E

General comments regarding use of CANDO.

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 using the Fletcher-Powell minimization algorithm requires the storing of an NB × NB matrix, it would, in general, be inconvenient to always dimension the above matrix 200 × 200.

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

```
DIMENSION HH(N, N), X(N), Y(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 CANDO are, approximately,

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

storage locations.

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

9. CANDO can handle zero valued R's, L's and C's as long as there are no loops of zero valued inductances or resistances (short circuits) and no cutsets of zero valued 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. Zero is not a valid branch or node number.

12. CANDO can be utilized 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.

Bibliography:

(1) R. Fletcher and M. J. D. Powell, "A rapidly convergent descent method for minimization," The Computer Journal, Vol. 6 (1963).