CANDOFD (Computer Analysis of Networks With Design Orientation in the Frequency Domain)

by

Paul M. Russo

Memorandum No. ERL-M253

9 September 1968

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

Table of Contents

Abstract:

This report describes the operation and use of CANDOFD (Computer Analysis of Networks with Design Orientation in the Frequency Domain), a frequency domain analysis program, for linear time invariant networks. The networks may contain dependent and independent sources of all types, capacitances, resistances and inductances.

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

In the program CANDOFD, nonindependent source tree voltages and nonindependent source link currents form a basis set of variables, denoted by $\underset{\wedge}{x}$. This formulation yields the automatic satisfaction of Kirchhoff's laws.

The problem reduces to solving a set of n coupled complex equations of the form

$$E_i(\underset{\wedge}{x}) = 0 \qquad (1)$$

The set of equations (1) is solved by minimizing a performance criterion ε, defined as

$$\varepsilon = \sum_{n} E_i \cdot E_i^{*} \qquad (2)$$

where * denotes the complex conjugate. Fletcher-Powell[2] and Rohrer[3] search minimization and step size selection algorithms are utilized, and the iterative procedure is terminated when ε is reduced to a desired value or when the desired number of minimization steps have taken place.

The tree picking and internal current scaling algorithms are such that large value spread and large impedance spread problems can be handled

reasonably effectively and efficiently. The DC response of a network may be obtained by simply solving the zero frequency problem.

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

Acknowledgments:

I wish to acknowledge the invaluable assistance and encouragement of Professor R. A. Rohrer, whose guidance was instrumental in the production of CANDOFD.

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:

CANDOFD is the frequency domain version of CANDO(1), and, as in CANDO, a trade-off exists between speed and accuracy of analysis. CANDOFD obtains a network analysis, at a specified frequency, by minimizing a performance criterion related to the complex branch errors, with Kirchhoff's laws being automatically (implicitly) satisfied. The iterative minimization scheme is terminated when either the squared branch error sum (performance criterion) falls below a desired value, or when a specified number of iterative minimization steps have been executed.

CANDOFD, with its accuracy vs speed trade off, is a Design Oriented analysis program in the sense that, in the initial phases of design, a quick (but not very accurate) analysis is desirable, whereas, in the final phase of design, an accurate analysis is necessary.

Section I describes the theory related to the formulation and minimization of the desired performance criterion, and indicates the solution to both the DC and general frequency point problems. Section II describes the MAIN program and subroutines associated with CANDOFD. The Appendices describe the use of CANDOFD in solving network problems, the way dependent sources must be modeled, and include a selection of sample problems.

in an open circuit, and no DC solution exists for such a network. If the independent current sources, in such a cutset, are zero valued, CANDOFD yields the correct network DC solution.

SECTION I: NETWORK ANALYSIS

Theory:

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

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

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

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

where $\underset{\sim}{Q}$ is the fundamental cutset matrix and $\underset{\sim}{B}$ is the fundamental loop matrix. Upon the selection of an optimal tree (see subroutine PT), we may renumber the NB branches of our network in such a way that the 1st NN-1 branches (NN is the number of nodes in our network), form the tree.

With the above numbering scheme, we may partition $\underset{\sim}{Q}$ and $\underset{\sim}{B}$ as follows:

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

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

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

and

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

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

Our optimal tree picking algorithm requires that all independent voltage sources be tree branches and that all independent current sources

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

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

and

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

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

To further reduce the order of the system, independent sources and reactive elements are not considered to be variables. (In the DC problem, all reactive elements act as zero valued independent sources.)

Thus the system of equations we are solving is one corresponding to a network consisting solely of independent sources (with real values), dependent sources and resistances.

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

<u>Solving the general frequency problem</u>:

We wish to find $\underset{\sim}{i}_b$ and $\underset{\sim}{v}_b$, the complex branch current and voltage vectors respectively, at a general frequency ω. The problem reduces to solving a coupled system of complex linear equations, consisting of the branch relations, and Kirchhoff's current and voltage laws.

A tree is selected according to our <u>optimal</u> 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 so that the 1st NN-1 branches form the tree, and the subsequent (NB-NN+1) branches are links. Kirchhoff's current and voltage laws become automatically satisfied by selecting the tree branch voltages and link currents as a basis set of variables.

If $\underset{\sim}{Q} = [\underset{\sim}{I} \mid \underset{\sim}{F}]$ is the fundamental cutset matrix associated with our optimal tree, we have

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

and

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

we thus wish to solve the complex set of coupled linear equations

$$\underset{\sim}{E}(\underset{\sim}{v}_t, \underset{\sim}{i}_\ell) = \underset{\sim}{0} \tag{1}$$

which are our complex branch errors (See Subroutine ERROR).

To further reduce the order of the system, independent source values are not treated as variables. Notice that this requires that independent voltage sources be tree branches, and that independent current sources be links.

The solution to (1) is obtained by minimizing a performance criterion ε, defined by

$$\varepsilon = \sum_i E_i^* \cdot E_i = \underset{\sim}{E}_i^* \cdot \underset{\sim}{E}_i \tag{2}$$

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

The Fletcher-Powell[2] minimization algorithm (see subroutine FLPOW) yields the directions along which ε is minimized, and Rohrer search[3] (see subroutine FNDALPH) is used to find the corresponding optimal step size.

The variables are selected to be the nonindependent source tree voltages (real and imaginary parts are distinct variables) and the nonindependent source link currents. The Fletcher-Powell and Rohrer search algorithms require the knowledge of the gradient vectors,

$$\frac{\partial \varepsilon}{\partial \underset{\wedge}{v}_{tr}}, \quad \frac{\partial \varepsilon}{\partial \underset{\wedge}{v}_{ti}}, \quad \frac{\partial \varepsilon}{\partial \underset{\wedge}{i}_{\ell r}}, \quad \frac{\partial \varepsilon}{\partial \underset{\wedge}{i}_{\ell i}}$$

where the subscripts r and i indicate the real and imaginary parts respectively. These gradient vectors are computed in subroutine FNDGRAD. Once ε is "sufficiently" minimized, $\underset{\wedge}{v}_t$ and $\underset{\wedge}{i}_\ell$ are "known" (to a desired accuracy). We then compute $\underset{\wedge}{i}_b$ and $\underset{\wedge}{v}_b$ as follows:

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

and

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

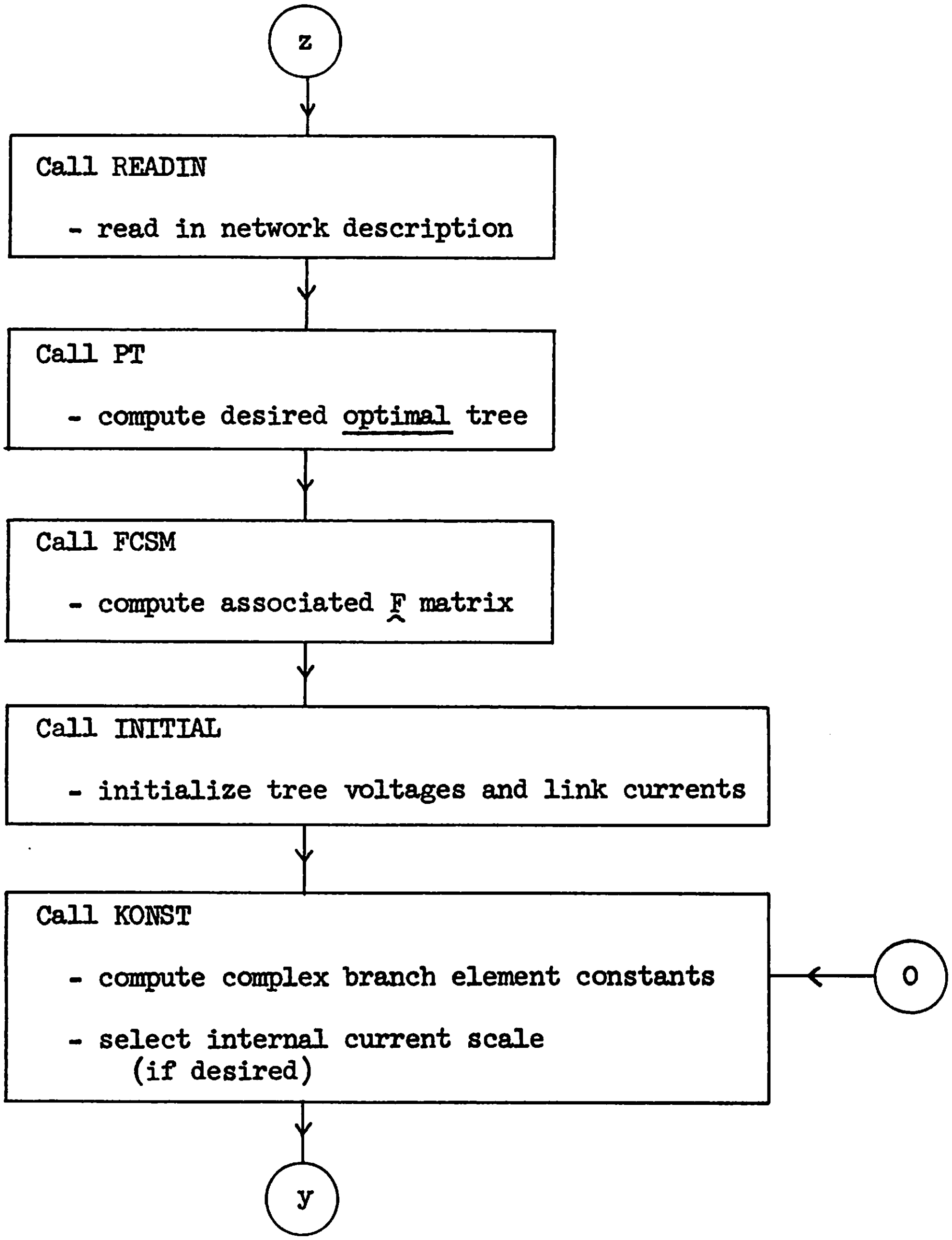
and hence, we proceed (if desired) to the next frequency point in an analogous way. The special case, for lossless networks, where the desired frequency corresponds to a resonance frequency, is discussed in the following section.

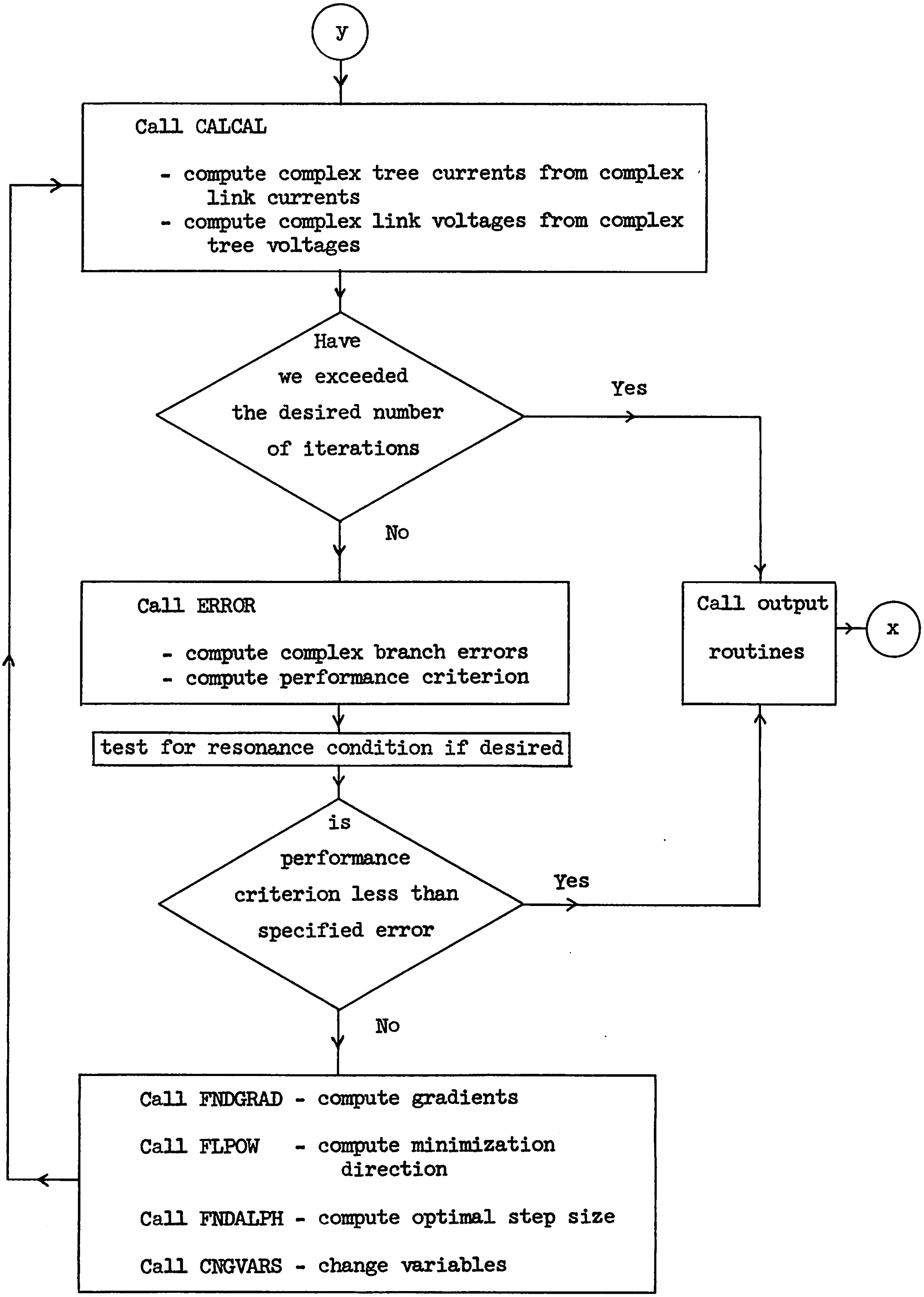
SECTION II. PROGRAM CANDOFD

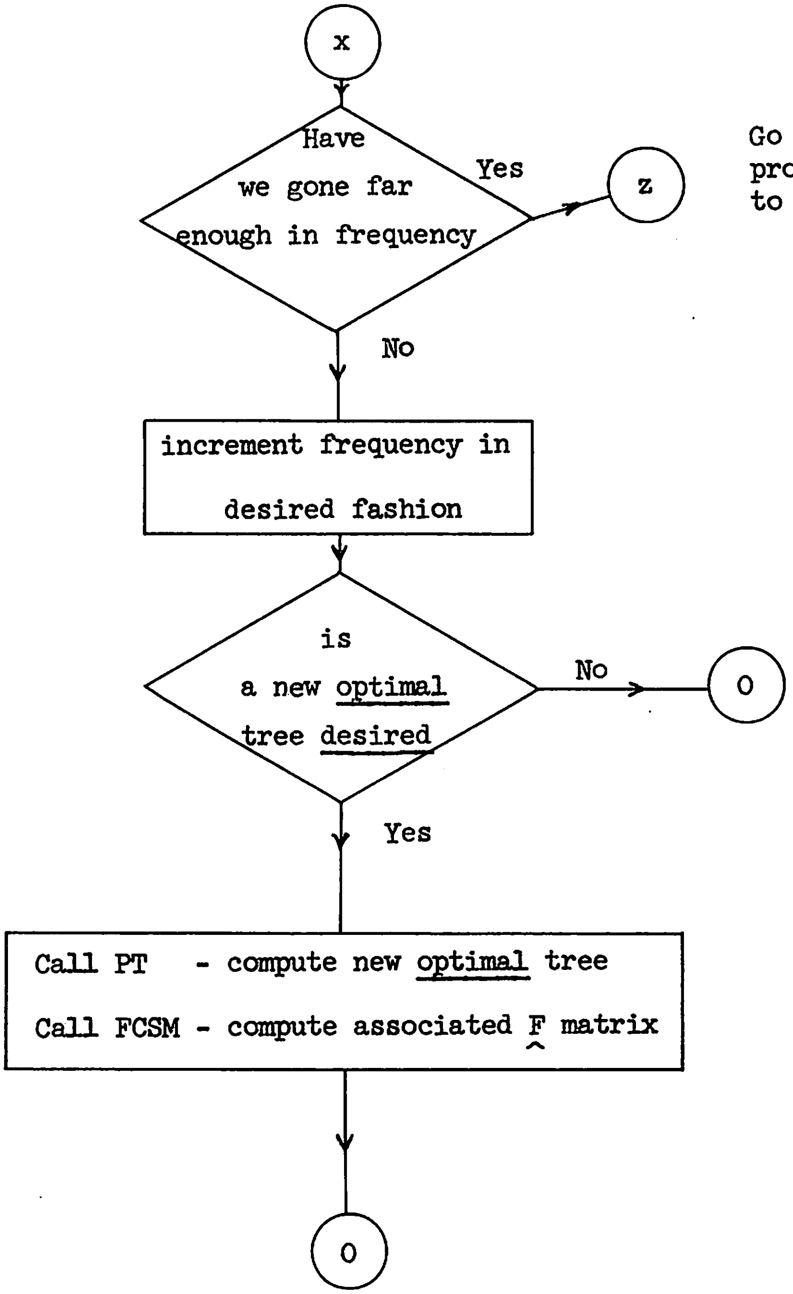
Main Program:

The main section of program CANDOFD ensures that the desired subroutines are called in the correct sequence. It solves the DC and general frequency point problems, in the way described in Section I. The main program also controls the calls to the output subroutines.

Block Diagram:


z
Call READIN
- read in network description
Call PT
- compute desired optimal tree
Call FCSM
- compute associated $\underset{\wedge}{F}$ matrix
Call INITIAL
- initialize tree voltages and link currents
Call KONST
- compute complex branch element constants
- select internal current scale (if desired)
o
y


y
Call CALCAL
- compute complex tree currents from complex link currents
- compute complex link voltages from complex tree voltages
Have we exceeded the desired number of iterations
Yes
No
Call ERROR
- compute complex branch errors
- compute performance criterion
Call output routines
x
test for resonance condition if desired
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 frequency
Yes
z
No
increment frequency in desired fashion
is a new optimal tree desired
No
O
Yes
Call PT - compute new optimal tree
Call FCSM - compute associated F̂ matrix
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 CANDOFD."

Subroutine PT:

At each call, subroutine PT computes an optimal tree. The optimal tree picking scheme will be enunciated below.

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

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 impedance magnitudes in ascending order of value

3 - independent current sources

The equivalent impedance magnitude values are given by the following prescription:

Controlled voltage source	→ 10^{-50}
Resistance (R)	→ R
Capacitance (C)	→ $\frac{1}{\omega * C}$
Inductance (L)	→ $\omega * L$
Controlled current source	→ 10^{50}

where ω is the frequency at which the optimal tree is being computed. When $\omega * C = 0$, the corresponding impedance magnitude is set to 10^{51}.

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 the presence of 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 set of data.

The internal numbering scheme is as follows:

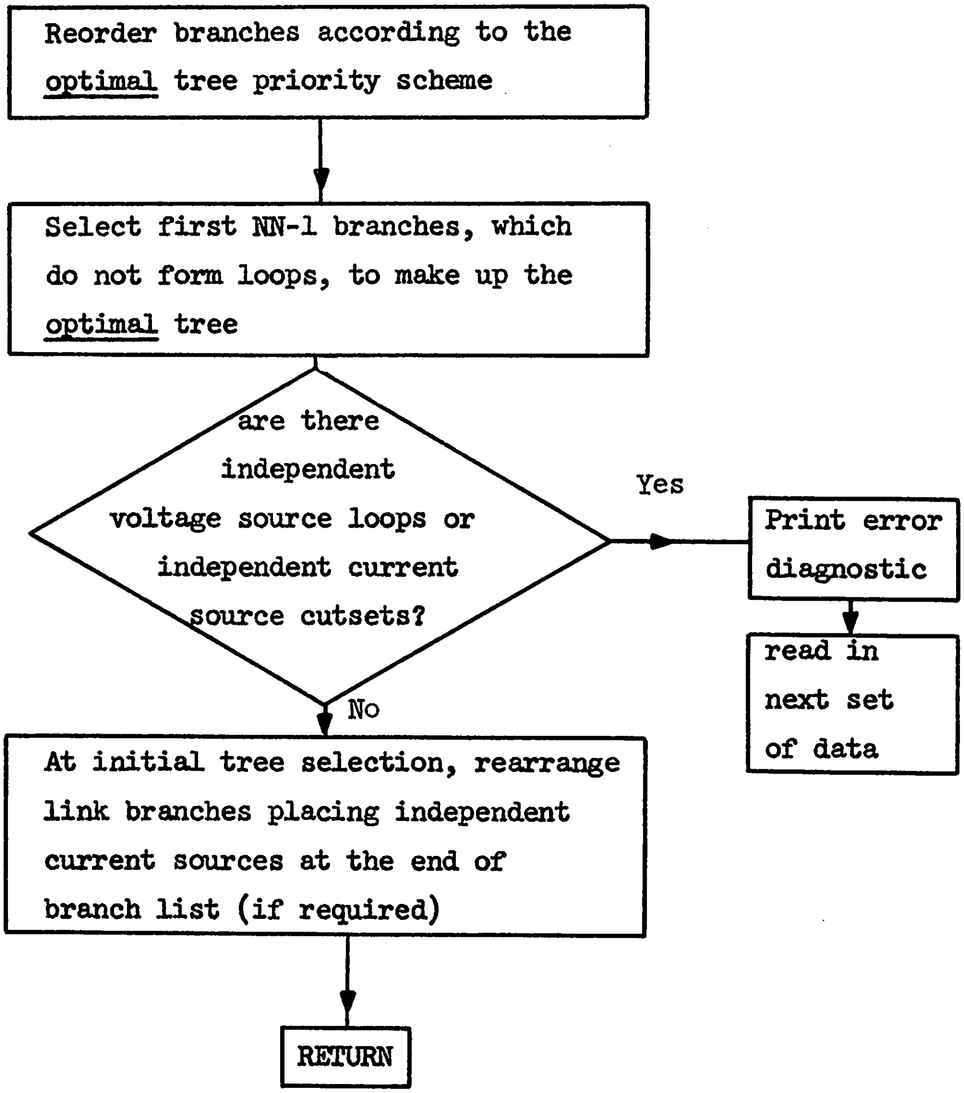
$$\left.\begin{array}{l}\left\{\begin{array}{l}\text{independent}\\ \text{voltage sources}\end{array}\right.\\ \\ \left\{\begin{array}{l}\text{remaining}\\ \text{tree branches}\end{array}\right.\end{array}\right\}\ \text{NN-1 tree branches}$$

$$\left.\begin{array}{l}\left\{\begin{array}{l}\text{remaining}\\ \text{links}\end{array}\right.\\ \\ \left\{\begin{array}{l}\text{independent}\\ \text{current sources}\end{array}\right.\end{array}\right\}\ \text{NB-NN+1 links}$$

The above numbering scheme allows us to ignore the independent source branches during subsequent optimal tree selections.

Subroutine PT, when desired, prints out the computed optimal tree and the rearranged network.

Block diagram:


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

Subroutine PT

Subroutine FCSM: (Fundamental Cutset Matrix)

Given a connected network, and a specified tree, there exists a unique (real) (NN-1 × NB) fundamental cutset matrix $\underset{\sim}{Q}$, 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

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

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

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

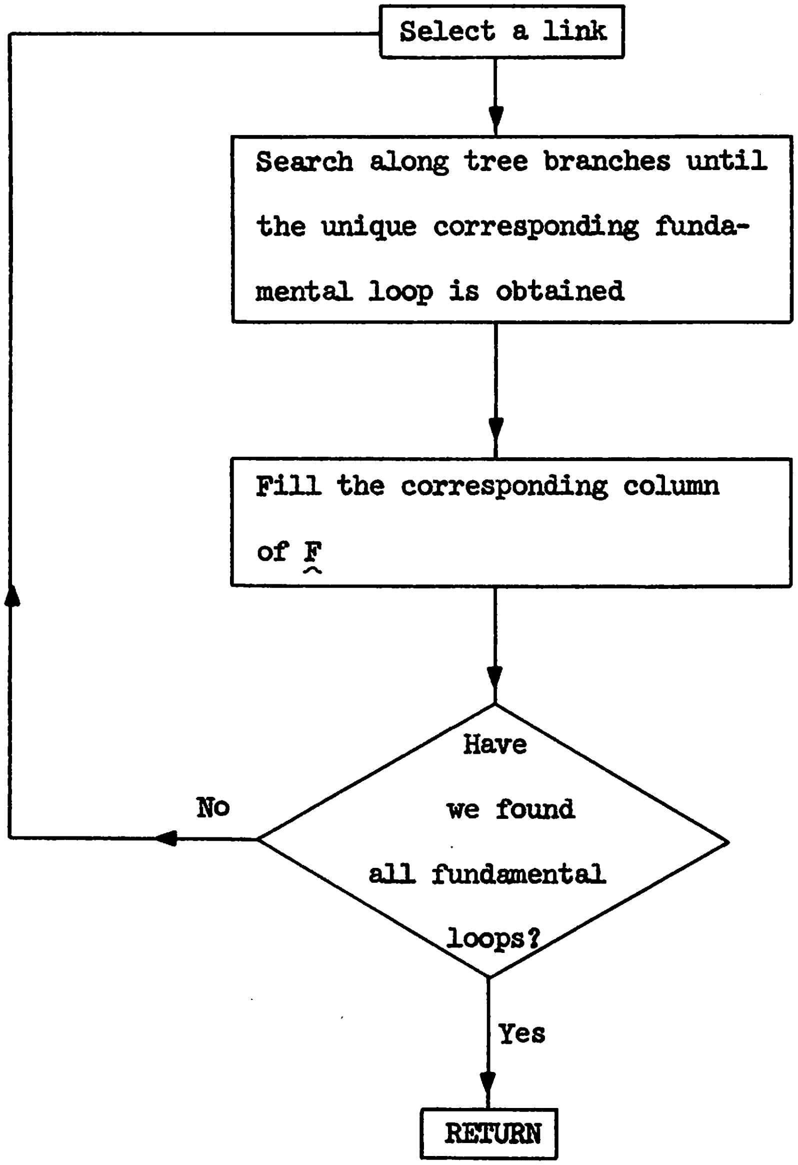
$$\underset{\sim}{Q} = [\underset{\sim}{I} \mid \underset{\sim}{F}] \quad ,$$

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

To each link, of our connected network, there corresponds a unique fundamental loop, consisting of the link itself and sufficient tree branches required to close the loop. Each column of $\underset{\sim}{F}$ corresponds to such a fundamental loop. This intermediate scheme is used to compute $\underset{\sim}{F}$.

If desired, FCSM can output the $\underset{\sim}{F}$ portion of the fundamental cutset matrix.

Block Diagram:


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

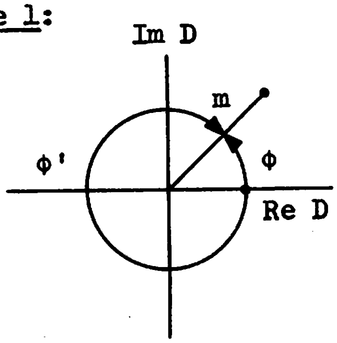
Subroutine FCSM

Subroutine VOLCUR:

This subroutine computes the real and imaginary components of independent source signals, from their magnitudes and phases (-360° to 360°).

Let D designate the complex independent source signal. Let m designate the given signal magnitude and let ϕ designate the given signal phase (-360° to 360°). ϕ' designates an equivalent, allowed, signal phase.

Case 1:



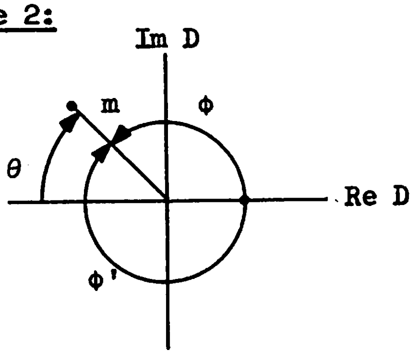
$$\text{Re D} = m \cdot \cos(\phi)$$

$$\text{Im D} = m \cdot \sin(\phi)$$

$$0 \leqq \phi \leqq 90^\circ$$

or $-360^\circ \leqq \phi' \leqq -270^\circ$

Case 2:



$$\text{Re D} = - m \cdot \cos(\theta)$$

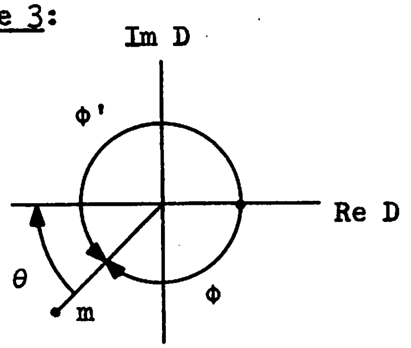
$$\text{Im D} = m \cdot \sin(\theta)$$

$$\theta \geqq 0$$

$$90^\circ \leqq \phi \leqq 180$$

or $-270^\circ \leqq \phi' \leqq -180^\circ$

Case 3:



$$\text{Re } D = -m \cdot \cos(\theta)$$

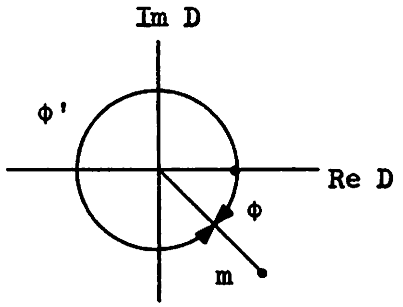
$$\text{Im } D = -m \cdot \sin(\theta)$$

$$\theta \geqq 0$$

$$180° \leqq \Phi' \leqq 270°$$

or $$-180° \leqq \Phi \leqq -90°$$

Case 4:



$$\text{Re } D = m \cdot \cos(\Phi)$$

$$\text{Im } D = -m \cdot \sin(\Phi)$$

$$270° \leqq \Phi' \leqq 360°$$

or $$-90° \leqq \Phi \leqq 0°$$

Independent source signals are read in in terms of magnitude and phase, for the sake of convenience. Subroutine VOLCUR transforms these signals into regular complex signals, which CANDOFD requires for execution.

Subroutine INITIAL:

This subroutine initializes the tree branch voltages and link currents of a given network. The independent source signals are set to their appropriate complex values, and the remaining signals are set equal to complex zeros. When $\omega = 0$, the independent source phases are ignored, and the real parts of the independence source signals are set equal to the signal magnitudes.

Subroutine CALCAL:

This subroutine computes the complex tree branch currents $\underset{\sim}{i}_b$ and the complex link voltages $\underset{\sim}{v}_\ell$ from the complex link currents $\underset{\sim}{i}_\ell$ and the complex 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 (and real) 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 connected network, by forcing $\underset{\sim}{i}_t$ and $\underset{\sim}{v}_\ell$ to satisfy the above relations.

SUBROUTINE KONST:

This subroutine computes the (complex) branch element constants required in the branch relations (see Subroutine ERROR). When desired (NSCALE $\neq 1$), KONST selects a new current scale factor each time a new topological tree is computed. There are no constants associated with independent sources.

The complex constants are computed according to the prescription given below, where ω is the source frequency, S denotes the current scale factor (e.g., $S = 1000 \Rightarrow$ currents are in milliamps) and λ_i denotes the constant associated with the ith branch.

Dependent voltage source:

a) Voltage controlled case

$$\mathrm{Re}(\lambda_i) = K_i$$

$$\mathrm{Im}(\lambda_i) = 0$$

where K_i is the ith coupling constant and is unitless.

b) Current controlled case

$$\mathrm{Re}(\lambda_i) = K_i/S$$

$$\mathrm{Im}(\lambda_i) = 0$$

where K_i is the ith coupling constant, in ohms.

Capacitances:

a) Tree branch case

$$\mathrm{Re}(\lambda_i) = 0$$

$$Im(\lambda_i) = \begin{cases} -\dfrac{1}{\omega * S * C_i} & \omega \neq 0 \\ \text{not defined} & \omega = 0 \end{cases}$$

b) Link case

$$Re(\lambda_i) = 0$$

$$Im(\lambda_i) = \omega * C_i * S$$

where C_i is the ith capacitance value, in farads,

Resistances:

a) Tree branch case

$$Re(\lambda_i) = R_i/S$$

$$Im(\lambda_i) = 0$$

b) Link case

$$Re(\lambda_i) = S/R_i$$

$$Im(\lambda_i) = 0$$

where R_i is the ith resistance value in ohms.

Inductances:

a) Tree branch case

$$Re(\lambda_i) = 0$$

$$Im(\lambda_i) = \frac{\omega * L_i}{S}$$

b) Link case

$$Re(\lambda_i) = 0$$

$$Im(\lambda_i) = \begin{cases} \dfrac{-S}{\omega * L_i} & \omega \neq 0 \\ \text{not defined} & \omega = 0 \end{cases}$$

where L_i is the ith inductance value in henrys.

Dependent current sources:

a) Voltage controlled case

$$Re(\lambda_i) = K_i * S$$

$$Im(\lambda_i) = 0$$

where K_i is the ith coupling constant, in ohms.

b) Current controlled case

$$Re(\lambda_i) = K_i$$

$$Im(\lambda_i) = 0$$

where K_i is the ith coupling constant, and is unitless.

It should be noted that no zero valued inductances can be links and no zero valued capacitances can be tree branches (see comment 8, Appendix E).

Automatic computation of scale factor:

When desired, subroutine KONST will compute a new current scale factor each time a new topological tree is selected.

The algorithm, for selecting the optimum scale factor, is as follows:

1. Select smallest link impedance magnitude (say $|Z_i|$).
2. Set internal current unit to be $\frac{\text{amps}}{|Z_i|}$, where $|Z_i|$ is in ohms, e.g., if $|Z_i| = 14800$ ohms, the internal current unit becomes $\frac{1}{14.8}$ milliamps.

If our optimal tree algorithm yields a topology which contains only link current sources, the scale factor is automatically set equal to 1.0.

SUBROUTINE ERROR:

This subroutine computes the complex branch errors E_i, and the performance criterion ε, defined by

$$\varepsilon = \frac{1}{2} \sum_i E_i \cdot E_i^*$$

where i ranges over all non-independent source branches and * designates the complex conjugate.

ε is to be subsequently minimized and, when both the branch relations and Kirchhoff's laws are simultaneously satisfied, ε will be identically zero.

Complex branch errors:

Notation:

The λ_i denote the complex branch element constants computed in subroutine KONST. V_i and i_i denote the complex branch voltage and current respectively. The E_i, of course, denote the complex branch error associated with the ith branch.

Dependent voltage sources:

a) Voltage controlled case

$$E_i = - V_i + \lambda_i * V_m$$

b) Current controlled case

$$E_i = - V_i + \lambda_i * i_m$$

where m denotes the controlling branch.

Capacitances:

a) Tree branch case.

$$E_i = -V_i + \lambda_i * i_i \qquad \omega \neq 0$$

$$E_i = \text{complex zero} \qquad \omega = 0$$

b) Link case

$$E_i = - i_i + \lambda_i * V_i$$

Resistances:

a) Tree branch case

$$E_i = - V_i + \lambda_i * i_i$$

b) Link case

$$E_i = - i_i + \lambda_i * V_i$$

Inductances:

a) Tree branch case

$$E_i = - V_i + \lambda_i * i_i$$

b) Link case

$$E_i = - i_i + \lambda_i * V_i \quad \omega \neq 0$$

$$E_i = \text{complex zero} \quad \omega = 0$$

Dependent current sources:

a) Voltage controlled case

$$E_i = - i_i + \lambda_i * V_m$$

b) Current controlled case

$$E_i = - i_i + \lambda_i * i_m$$

where m denotes the controlling branch.

SUBROUTINE CNGVARS:

Given the complex direction along which we minimized, $\underset{\sim}{s}^i$, and the (real) optimal step size α^i, the complex non independent source tree voltages and link currents are modified as follows:

$$\begin{bmatrix} \hat{\underset{\sim}{v}}_t \\ --- \\ \hat{\underset{\sim}{i}}_\ell \end{bmatrix}_{new} = \begin{bmatrix} \hat{\underset{\sim}{v}}_t \\ --- \\ \hat{\underset{\sim}{i}}_\ell \end{bmatrix}_{old} - \alpha^i * \underset{\sim}{s}^i$$

It should be emphasized that only the basis set of variables, i.e., the non independent source tree voltages and link currents, are changed. $\hat{\underset{\sim}{v}}_t$ and $\hat{\underset{\sim}{i}}_\ell$ designate, in this context, non-independent source signals. Tree currents and link voltages follow directly from the implicit formulation of Kirchhoff's laws (see subroutine CALCAL).

SUBROUTINE FNDGRAD:

This subroutine computes the complex gradient vector of the performance criterion, with respect to non-independent source tree voltages and non-independent source link currents.

Theory:

a) Tree voltage gradient vector

define $$\mathrm{Re}(\underset{\sim}{g}_t) = \frac{d\varepsilon}{d[\mathrm{Re}\underset{\sim}{v}_t]}$$

and $$\mathrm{Im}(\underset{\sim}{g}_t) = \frac{d\varepsilon}{d[\mathrm{Im}\underset{\sim}{v}_t]} \qquad (1)$$

where $\underset{\sim}{g}_t$ is the tree voltage complex gradient vector.

Recalling that

$$\underset{\sim}{v}_b = \begin{bmatrix} \underset{\sim}{I} \\ \hline \underset{\sim}{F}' \end{bmatrix} \underset{\sim}{v}_t \qquad (2)$$

the tree voltage gradient vector $\underset{\sim}{g}_t$ can be expressed, via the chain rule, as

$$\underset{\sim}{g}_t = \left[\frac{\partial \underset{\sim}{v}_b}{\partial \underset{\sim}{v}_t}\right] \cdot \frac{\partial \varepsilon}{\partial \underset{\sim}{v}_b} \qquad (3)$$

where

$$\frac{\partial \underset{\sim}{v}_b}{\partial \underset{\sim}{v}_t} = \begin{bmatrix} \frac{\partial v_{b_1}}{\partial v_{t_1}} & \frac{\partial v_{b_2}}{\partial v_{t_1}} & \cdots\cdots & \frac{\partial v_{b_{NB}}}{\partial v_{t_1}} \\ & & & \\ \frac{\partial v_{b_1}}{\partial v_{t_{NN-1}}} & \cdots\cdots & \cdots & \frac{\partial v_{b_{NB}}}{\partial v_{t_{NN-1}}} \end{bmatrix} \qquad (4)$$

also

$$\left.\begin{aligned} &\mathrm{Re}\left[\frac{\partial\varepsilon}{\partial \underset{\sim}{v}_b}\right] = \frac{\partial\varepsilon}{\partial[\mathrm{Re}\,\underset{\sim}{v}_b]} \\ \text{and}\qquad &\mathrm{Im}\left[\frac{\partial\varepsilon}{\partial \underset{\sim}{v}_b}\right] = \frac{\partial\varepsilon}{\partial[\mathrm{Im}\,\underset{\sim}{v}_b]} \end{aligned}\right\} \tag{5}$$

from (2), it is immediately apparent that

$$\frac{\partial \underset{\sim}{v}_b}{\partial \underset{\sim}{v}_t} = \begin{bmatrix} \underset{\sim}{I} \\ \hline \underset{\sim}{F}' \end{bmatrix} \tag{6}$$

Now, $\frac{\partial\varepsilon}{\partial \underset{\sim}{v}_b}$ may be partitioned as follows:

$$\frac{\partial\varepsilon}{\partial \underset{\sim}{v}_b} = \begin{bmatrix} \frac{\partial\varepsilon}{\partial \underset{\sim}{v}_t} \\ \hline \frac{\partial\varepsilon}{\partial \underset{\sim}{v}_\ell} \end{bmatrix} \tag{7}$$

Combining (3), (6) and (7) we obtain

$$\underset{\sim}{g}_t = [\mathrm{I} : \underset{\sim}{F}] \cdot \begin{bmatrix} \frac{\partial\varepsilon}{\partial \underset{\sim}{v}_t} \\ \hline \frac{\partial\varepsilon}{\partial \underset{\sim}{v}_\ell} \end{bmatrix} \tag{8}$$

Equation (8) immediately reduces to the desired form

$$\underset{\sim}{g}_t = \frac{d\varepsilon}{d\underset{\sim}{v}_t} = \frac{\partial\varepsilon}{\partial \underset{\sim}{v}_t} + \underset{\sim}{F} \cdot \frac{\partial\varepsilon}{\partial \underset{\sim}{v}_\ell} \tag{9}$$

where $$\text{Re}\left[\frac{\partial\,\varepsilon}{\partial \underset{\sim}{v}_t}\right] = \frac{\partial\varepsilon}{\partial[\text{Re}\underset{\sim}{v}_t]} \quad ,$$

$$\text{Im}\left[\frac{\partial\varepsilon}{\partial v_t}\right] = \frac{\partial\varepsilon}{\partial[\text{Im}\underset{\sim}{v}_t]} \quad ,$$

$$\text{Re}\left[\frac{\partial\varepsilon}{\partial \underset{\sim}{v}_\ell}\right] = \frac{\partial\varepsilon}{\partial[\text{Re}\underset{\sim}{v}_\ell]} \quad ,$$

and $$\text{Im}\left[\frac{\partial\varepsilon}{\partial \underset{\sim}{v}_\ell}\right] = \frac{\partial\varepsilon}{\partial[\text{Im}\underset{\sim}{v}_\ell]} \quad .$$

b) Link current gradient vector.

Since

$$\underset{\sim}{i}_b = \begin{bmatrix} -\underset{\sim}{F} \\ \text{----} \\ \underset{\sim}{I} \end{bmatrix} \cdot \underset{\sim}{i}_\ell \qquad (10)$$

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

$$\underset{\sim}{g}_\ell = \frac{d\varepsilon}{d\underset{\sim}{i}_\ell} = \frac{\partial\varepsilon}{\partial\underset{\sim}{i}_\ell} - \underset{\sim}{F}' \cdot \frac{\partial\varepsilon}{\partial\underset{\sim}{i}_t} \qquad (11)$$

where $\underset{\sim}{g}_\ell$, $\frac{d\varepsilon}{d\underset{\sim}{i}_\ell}$, $\frac{\partial\varepsilon}{\partial\underset{\sim}{i}_\ell}$ and $\frac{\partial\varepsilon}{\partial\underset{\sim}{i}_t}$ have analogous meanings to their counterparts in the tree voltage gradient case.

c) Branch gradient vectors.

The branch gradients $\frac{\partial\varepsilon}{\partial\underset{\sim}{i}_b}$ and $\frac{\partial\varepsilon}{\partial\underset{\sim}{v}_b}$ are computed as follows; (where $\frac{\partial\varepsilon}{\partial v_i}$ and $\frac{\partial\varepsilon}{\partial i_i}$ are branch gradients associated with the ith branch):

Tree branches:

Independent voltage sources:

a) not associated with controlled sources $\frac{\partial\varepsilon}{\partial v_i} = \frac{\partial\varepsilon}{\partial i_i}$ = complex zero

b) associated with controlled source m

$$\frac{\partial \varepsilon}{\partial v_i} = \text{complex zero}$$

$$\frac{\partial \varepsilon}{\partial i_i} = \lambda_m * E_m$$

- dependent voltage sources:

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

$$\frac{\partial \varepsilon}{\partial i_i} = \text{complex zero}$$

- dependent current sources:

$$\frac{\partial \varepsilon}{\partial v_i} = \text{complex zero}$$

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

- tree capacitances at $\omega = 0$:

$$\frac{\partial \varepsilon}{\partial v_i} = \frac{\partial \varepsilon}{\partial i_i} = \text{complex zero}$$

- remaining tree branches:

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

$$\text{Re}\left[\frac{\partial \varepsilon}{\partial i_i}\right] = \text{Re}(\lambda_i) * \text{Re}(E_i) + \text{Im}(\lambda_i) * \text{Im}(E_i)$$

$$\text{Im}\left[\frac{\partial \varepsilon}{\partial i_i}\right] = - \text{Im}(\lambda_i) * \text{Re}(E_i) + \text{Re}(\lambda_i) * \text{Im}(E_i)$$

<u>Link branches</u>:

- Independent current sources:

a) not associated with controlled sources

$$\frac{\partial \varepsilon}{\partial v_i} = \frac{\partial \varepsilon}{\partial i_i} = \text{complex zero}$$

b) associated with controlled source m

$$\frac{\partial \varepsilon}{\partial v_i} = \lambda_m * E_m$$

$$\frac{\partial \varepsilon}{\partial i_i} = \text{complex zero}$$

- dependent current sources:

$$\frac{\partial \varepsilon}{\partial v_i} = \text{complex zero}$$

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

- dependent voltage sources:

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

$$\frac{\partial \varepsilon}{\partial i_i} = \text{complex zero}$$

- link inductance at $\omega = 0$:

$$\frac{\partial \varepsilon}{\partial v_i} = \frac{\partial \varepsilon}{\partial i_i} = \text{complex zero}$$

- remaining links

$$\text{Re}\left[\frac{\partial \varepsilon}{\partial v_i}\right] = \text{Re}(\lambda_i) * \text{Re}(E_i) + \text{Im}(\lambda_i) * \text{Im}(E_i)$$

$$\text{Im}\left[\frac{\partial \varepsilon}{\partial v_i}\right] = - \text{Im}(\lambda_i) * \text{Re}(E_i) + \text{Re}(\lambda_i) * \text{Im}(E_i)$$

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

Hence, knowing $\frac{\partial \varepsilon}{\partial \underset{\sim}{i}_b}$ and $\frac{\partial \varepsilon}{\partial \underset{\sim}{v}_b}$, $\frac{d\varepsilon}{d\underset{\sim}{v}_t}$ and $\frac{d\varepsilon}{d\underset{\sim}{i}_\ell}$ follow directly from equations (9) and (11). Note that, at $\omega = 0$, $\frac{d\varepsilon}{dv}$ for tree capacitances and $\frac{d\varepsilon}{di}$ for link inductances must be constrained to be complex zero.

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 (complex) direction computed by subroutine FLPOW.

The constant α^i is such that

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

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

The analytical expression for α^i is:

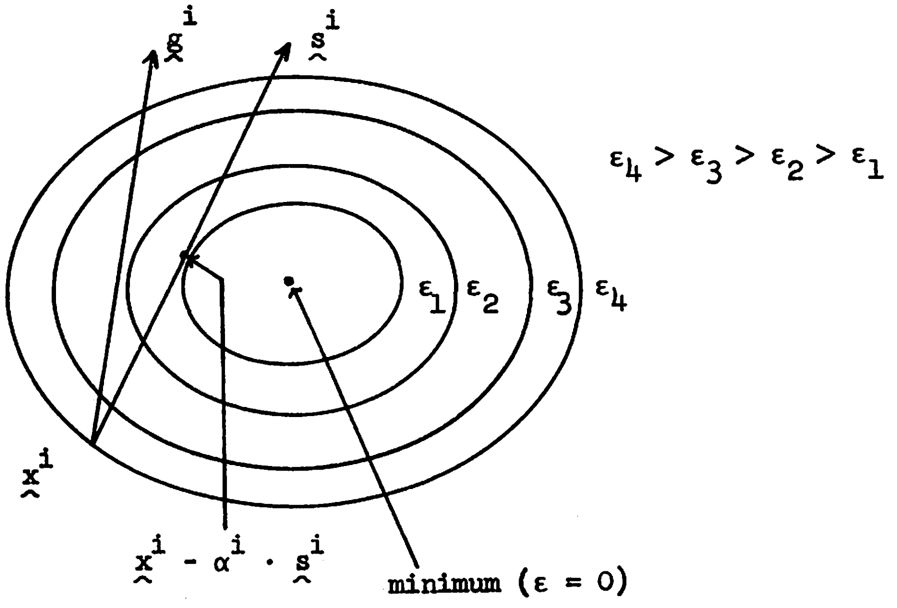
$$\alpha^i = \frac{1}{2} \frac{A}{\varepsilon(\underset{\wedge}{x} - \underset{\wedge}{S}^i) + A - \varepsilon(\underset{\wedge}{x})} ,$$

and

$$A = \sum_k [\mathrm{Re}(g_k^i) \cdot \mathrm{Re}(S_k^i) + \mathrm{Im}(g_k^i) \cdot \mathrm{Im}(S_k^i)]$$

where $\underset{\wedge}{g}^i$ is the complex gradient at the ith iterative step, and k ranges over all the nonindependent source branches.

The situation may be graphically depicted as follows:



$\underset{\wedge}{g}^i$
$\underset{\wedge}{s}^i$
$\varepsilon_4 > \varepsilon_3 > \varepsilon_2 > \varepsilon_1$
ε_1
ε_2
ε_3
ε_4
$\underset{\wedge}{x}^i$
$\underset{\wedge}{x}^i - \alpha^i \cdot \underset{\wedge}{s}^i$
minimum (ε = 0)

SUBROUTINE FLPOW:

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

The Fletcher-Powell(2) 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 dimension of $\underset{\wedge}{H}$ is N × N where

N = 2× (NB - number of independent sources)

The direction along which we perform the minimization is given by

$$\underset{\wedge}{y}^{i} = \underset{\wedge}{H}^{i} \cdot \underset{\wedge}{\omega}^{i} \quad ,$$

where the superscript i indicates that we are at the ith iterative step, and $\underset{\wedge}{\omega}^{i}$ is defined as follows:

$$\underset{\wedge}{\omega}^{i} = \begin{bmatrix} \text{Re } (\underset{\sim}{g}_t) \\ \text{Re } (\underset{\sim}{g}_\ell) \\ \text{Im } (\underset{\sim}{g}_t) \\ \text{Im } (\underset{\sim}{g}_\ell) \end{bmatrix}$$

where $\underset{\sim}{g}_t$ and $\underset{\sim}{g}_\ell$ are defined in subroutine FNDGRAD.

The $\underset{\wedge}{H}$ matrix is updated as follows:

$$\underset{\wedge}{H}^{1} = \underset{\wedge}{I} \text{ (Identity matrix)}$$

$$\underset{\wedge}{H}^{i+1} = \underset{\wedge}{H}^{i} - \frac{\underset{\wedge}{y}^{i} \cdot (\underset{\wedge}{y}^{i})'}{(\underset{\wedge}{y}^{i})' \cdot (\underset{\wedge}{\omega}^{i+1} - \underset{\wedge}{\omega}^{i})} - \frac{[\underset{\wedge}{H}^{i} \cdot (\underset{\wedge}{\omega}^{i+1} - \underset{\wedge}{\omega}^{i})] \cdot [\underset{\wedge}{H}^{i} \cdot (\underset{\wedge}{\omega}^{i+1} - \underset{\wedge}{\omega}^{i})]'}{(\underset{\wedge}{\omega}^{i+1} - \underset{\wedge}{\omega}^{i})' \cdot \underset{\wedge}{H}^{i} \cdot (\underset{\wedge}{\omega}^{i+1} - \underset{\wedge}{\omega}^{i})}$$

Then,

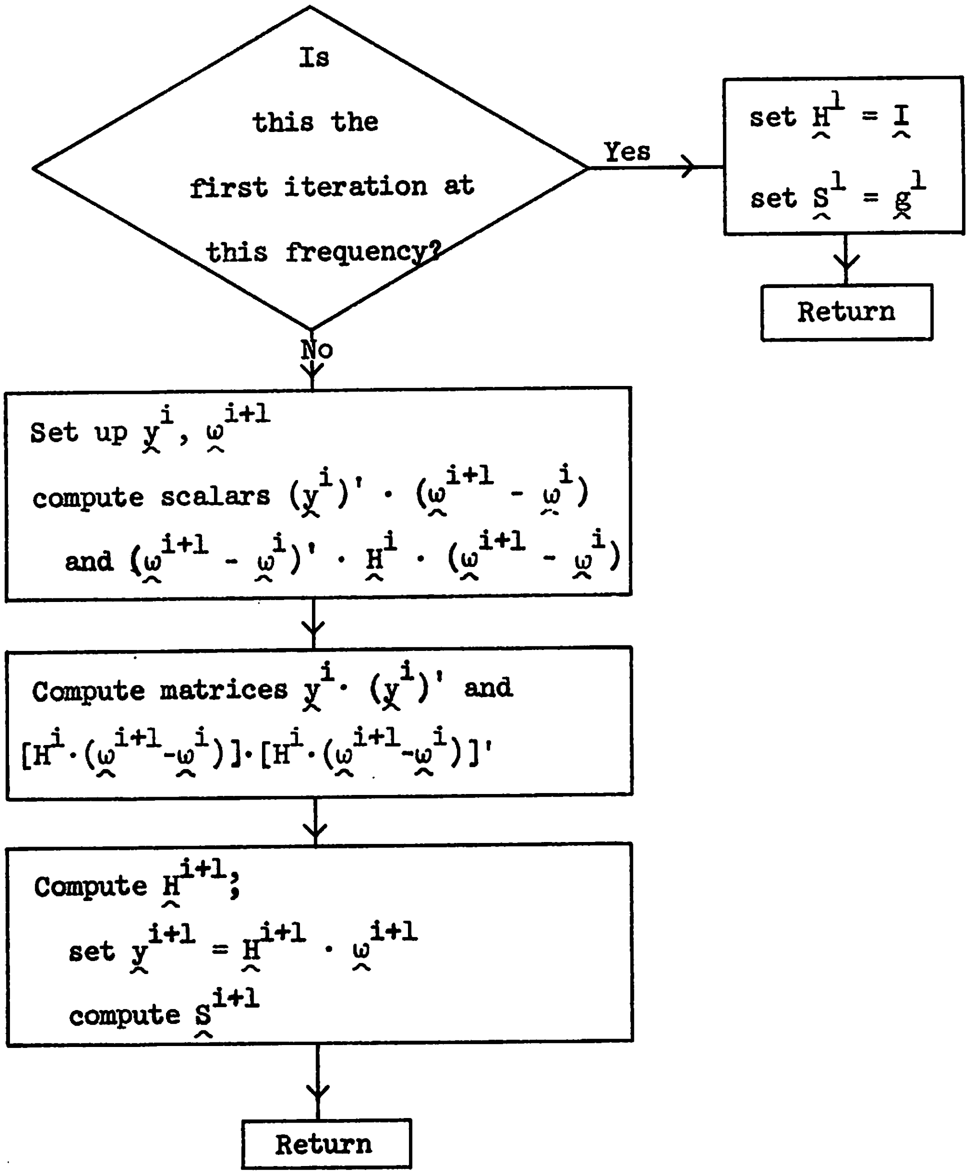
$$\underset{\wedge}{y}^{i+1} = \underset{\wedge}{H}^{i+1} \cdot \underset{\wedge}{\omega}^{i+1}$$

and, we identify

$$\begin{bmatrix} \mathrm{Re}\ (\underset{\wedge}{S}^{i+1}) \\ \text{---------} \\ \mathrm{Im}\ (\underset{\wedge}{S}^{i+1}) \end{bmatrix} = \underset{\wedge}{y}^{i+1}$$

The $\underset{\wedge}{H}$ matrix converges to the inverse of the second derivative matrix, known as the Hessian matrix.

Block Diagram


Is this the first iteration at this frequency?
Yes
set $\underset{\wedge}{H}^1 = \underset{\wedge}{I}$
set $\underset{\wedge}{S}^1 = \underset{\sim}{g}^1$
Return
No
Set up $\underset{\wedge}{y}^i$, $\underset{\wedge}{\omega}^{i+1}$
compute scalars $(\underset{\sim}{y}^i)' \cdot (\underset{\wedge}{\omega}^{i+1} - \underset{\wedge}{\omega}^i)$
and $(\underset{\wedge}{\omega}^{i+1} - \underset{\wedge}{\omega}^i)' \cdot \underset{\wedge}{H}^i \cdot (\underset{\wedge}{\omega}^{i+1} - \underset{\wedge}{\omega}^i)$
Compute matrices $\underset{\sim}{y}^i \cdot (\underset{\sim}{y}^i)'$ and
$[H^i \cdot (\underset{\wedge}{\omega}^{i+1} - \underset{\wedge}{\omega}^i)] \cdot [H^i \cdot (\underset{\wedge}{\omega}^{i+1} - \underset{\wedge}{\omega}^i)]'$
Compute $\underset{\wedge}{H}^{i+1}$;
set $\underset{\wedge}{y}^{i+1} = \underset{\wedge}{H}^{i+1} \cdot \underset{\wedge}{\omega}^{i+1}$
compute $\underset{\wedge}{S}^{i+1}$
Return

SUBROUTINE FLPOW

SUBROUTINE READOUT:

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

If the frequency at which the network was analyzed had to be modified due to the presence of an "ideal resonance," a diagnostic is provided.

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

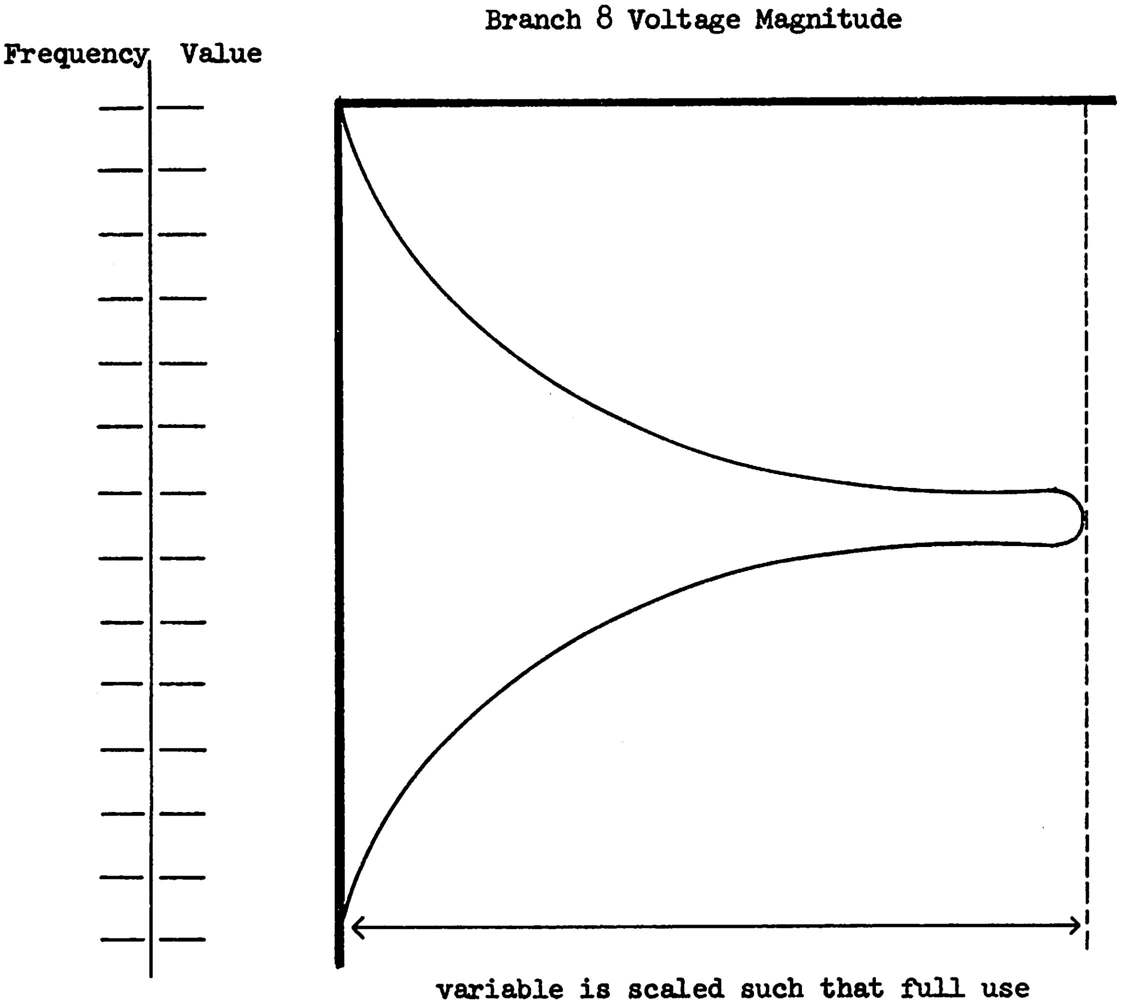
Subroutine READOUT converts the desired output variables from their internal complex representation, to a more desirable (magnitude, phase) representation.

SUBROUTINE GRAPH:

The purpose of this subroutine is to provide graphical outputs of up to five signals. A graphical output of any branch voltage or current, magnitude or phase, can be obtained.

Subroutine GRAPH is called, by the main program, at each frequency point. The desired voltages and currents are stored in an array. When either the final frequency point or the 200th frequency point is reached, the desired magnitudes and phases are computed. The variables are then scaled individually so that maximum use is made of the output page width.

The following figure illustrates a typical graph output.



variable is scaled such that full use
is made of page width.

SUBROUTINE ALLOUT:

This subroutine outputs all the tree voltages and link currents (including those associated with independent sources), in the optimal tree numbering scheme.

Hence, when ALLOUT is used, one should also output the corresponding tree information, which will immediately yield the isomorphism between the original network and the new topologies.

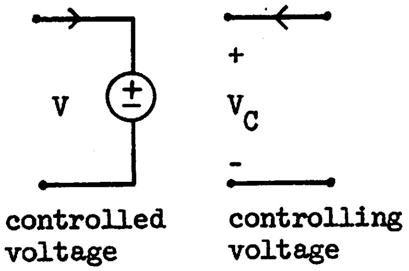
APPENDIX A:

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.

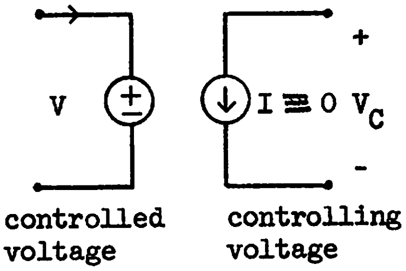
Voltage controlled voltage source:

Ideal model:



$V \equiv k\ V_C$

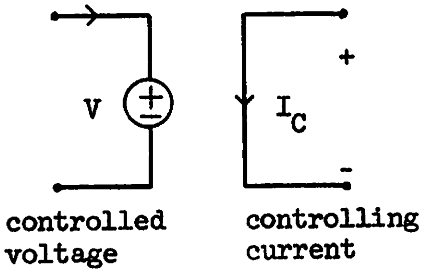
CANDOFD model:



$V \equiv k\ V_C$

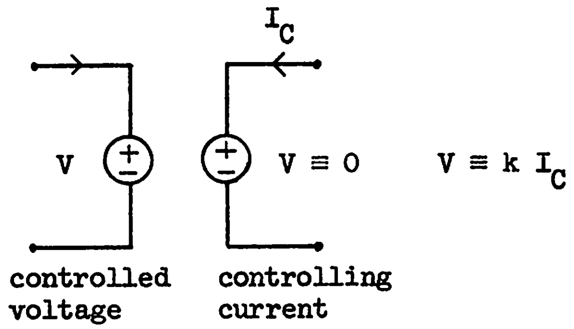
Current controlled voltage source:

Ideal model:



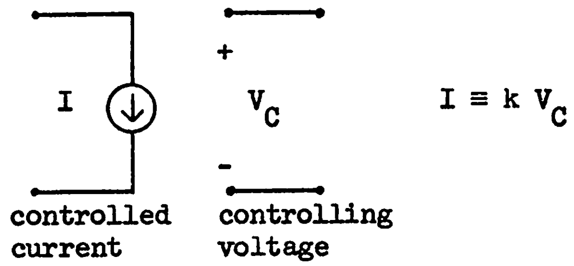
$V \equiv k\ I_C$

CANDOFD model:

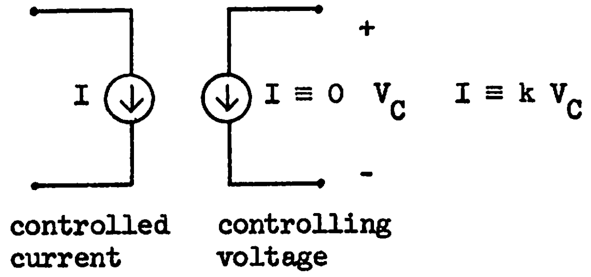


<u>Voltage controlled current source</u>:

Ideal model:

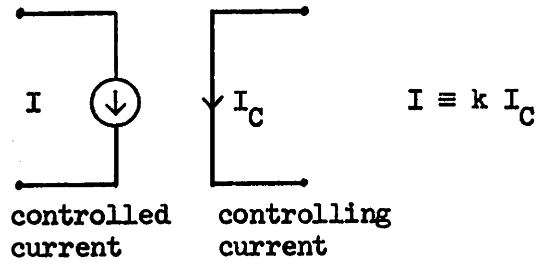


CANDOFD model:

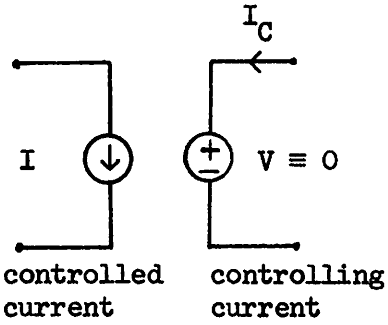


<u>Current controlled current source</u>:

Ideal model:



CANDOFD model:



$I \equiv k\ I_C$

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

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

APPENDIX B:

Data cards for CANDOFD

Notation:

I ⇒ integer format

A ⇒ alphanumeric format

E ⇒ exponential or floating point format

col ⇒ column on data card

Card #1:

Variables read in, in sequential order

NN, number of nodes (I)

NB, number of branches (I)

WSTART, initial frequency (E)

WEND, final frequency (E)

FACTOR, frequency multiplicative factor, if desired, (E)

EPS, error below which performance criterion must fall (E)

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

NCONT = 1 ⇒ outputs desired

NCONT = 0 ⇒ outputs not desired

NTREE, Tree picking control (I)

NTREE = 1 ⇒ pick a new tree at each frequency point

NTREE = 0 ⇒ pick a tree only at the initial frequency point

col 1-5,	- NN	DATA CARD #1
col 6-10,	- NB	
col 11-25,	- WSTART	
col 26-40,	- WEND	
col 41-55,	- FACTOR	
col 56-70,	- EPS	
col 75,	- NCONT	
col 80,	- NTREE	

Card #2:

Variables read in, in sequential order

NGRAPH, number of graphical outputs (I)

NALLOUT, control variable for use of

ALLOUT subroutine (I)

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

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

JOUT, number of outputs desired (I)

To be used in conjunction with subroutine READOUT

SCALE, scale factor (E)

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

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

If only an error control is desirable, set NITT = 0

FREQADD, frequency additive step, if desired (E)

NSCALE, Control variable for use of internal, automatic current scaling

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

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

col 1-5,	- NGRAPH	DATA CARD #2
col 6-10,	- NALLOUT	
col 11-15,	- JOUT	
col 16-30,	- SCALE	
col 31-35,	- NITT	
col 36-50,	- FREQADD	
col 51-55,	- NSCALE	
col 56-60,	- IDRES	

IDRES, Control variable for use of internal ideal resonance testing.

IDRES = 1 $\Rightarrow$ bypass ideal resonance tests

IDRES $\neq$ 1 $\Rightarrow$ test for ideal resonances

Network description data cards:

For each network branch, the following card 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)

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)

VALUE, value of branch element (E)

- Resistance in ohms

- Inductance in henrys

- Capacitance in farads

- Dependent source coupling constants in ohms, mhos, or unitless

(See Appendix A)

AMAG, independent source signal magnitude (E)

in volts, for independent voltage source

in amperes, for independent current source

APHASE, independent source signal phase, in degrees (E)

- 360° ≤ phase ≤ +360

col 1,	- TYPE	DATA CARD DESCRIBING ONE NETWORK BRANCH
col 2-4,	- IBRAN	
col 6,	- CONTYPE	
col 7-9,	- KONBRAN	
col 11-12,	- LEAV	
col 14-15,	- LENT	
col 21-35,	- VALUE	
col 36-50,	- AMAG	
col 51-65,	- APHASE	

<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 ≠ 0, we need NGRAPH (≤ 5) data cards with the following information

col 1-5, branch number (I)

col 10, branch signal desired (I)

1 ⇒ current desired

0 ⇒ voltage desired

col 15, signal information desired (I)

1 ⇒ phase plot desired

0 ⇒ magnitude plot desired

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

col 1-5, branch number (I)

col 10, branch signal desired (I)

1 ⇒ current desired

0 ⇒ 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:

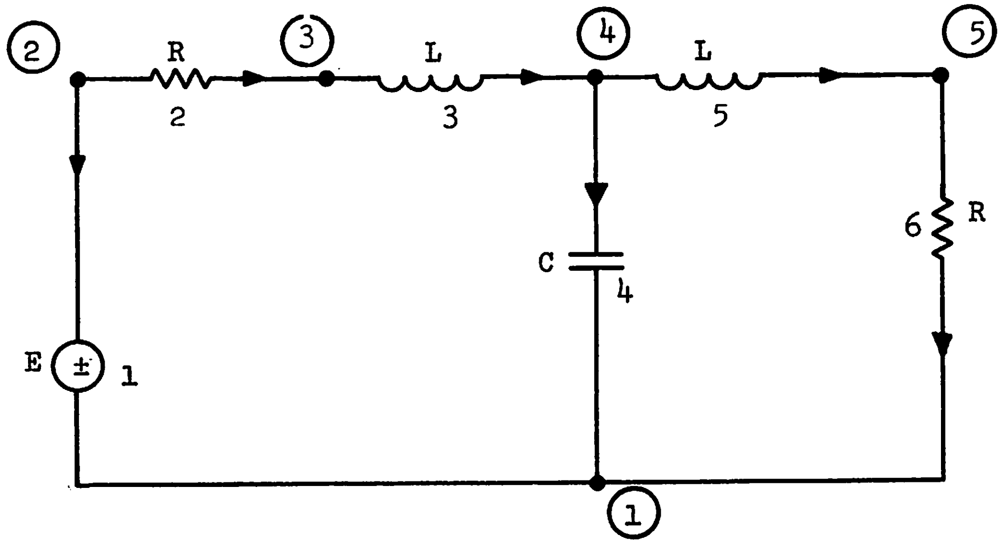
Sample Problems:

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
- WSTART is the initial frequency in radians/sec, (WSTART = 0 implies a DC solution is desired)
- WEND is the final frequency of concern (in radians/sec)
- FACTOR is the multiplicative frequency factor (when desired) e.g., if ω_1 is the present frequency point, the next frequency point will be given by ω_1 * FACTOR
- FREQADD is the additive frequency increment (when desired) e.g., if ω_1 is the present frequency point, the next frequency point is given by ω_1 + FREQADD
- EPS is the desired performance criterion error
- The central processor time, given for the sample problems, includes the input-output execution time.
- Only the pertinent computer output is reproduced with the sample problems .

Sample Problem #1:

Low Pass Butterworth Filter:



NN = 5

NB = 6

R = 10^3

L = 0.01

C = 2×10^{-8}

$|E| = 10$ V, ∢ E = 0.0°

Specifications:

- Initial frequency = 10^3 (WSTART = 10^3)
- Final frequency = 5×10^5 (WEND = 5×10^5)
- Frequency additive factor = 10^4 (FREQADD = 10^4)
- Desire error to fall below 10^{-5} or a maximum of 18 iterative minimization steps at each frequency point.

  (EPS = 10^{-5}, NITT = 18)

- Desire tree and $\underset{\wedge}{F}$ to be outputed (NCONT = 1)
- Desire internal automatic current scaling. (SCALE is arbitrary,

  NSCALE $\neq$ 1
- Desire the following graphical outputs

$$\left.\begin{array}{l}\text{Branch 4 current magnitude}\\ \text{Branch 1 current magnitude}\\ \text{Branch 1 current phase}\\ \text{Branch 6 voltage magnitude}\\ \text{Branch 6 voltage phase}\end{array}\right\}\qquad (\text{NGRAPH} = 5)$$

- Desire an optimal tree to be selected at each frequency (NTREE = 1)

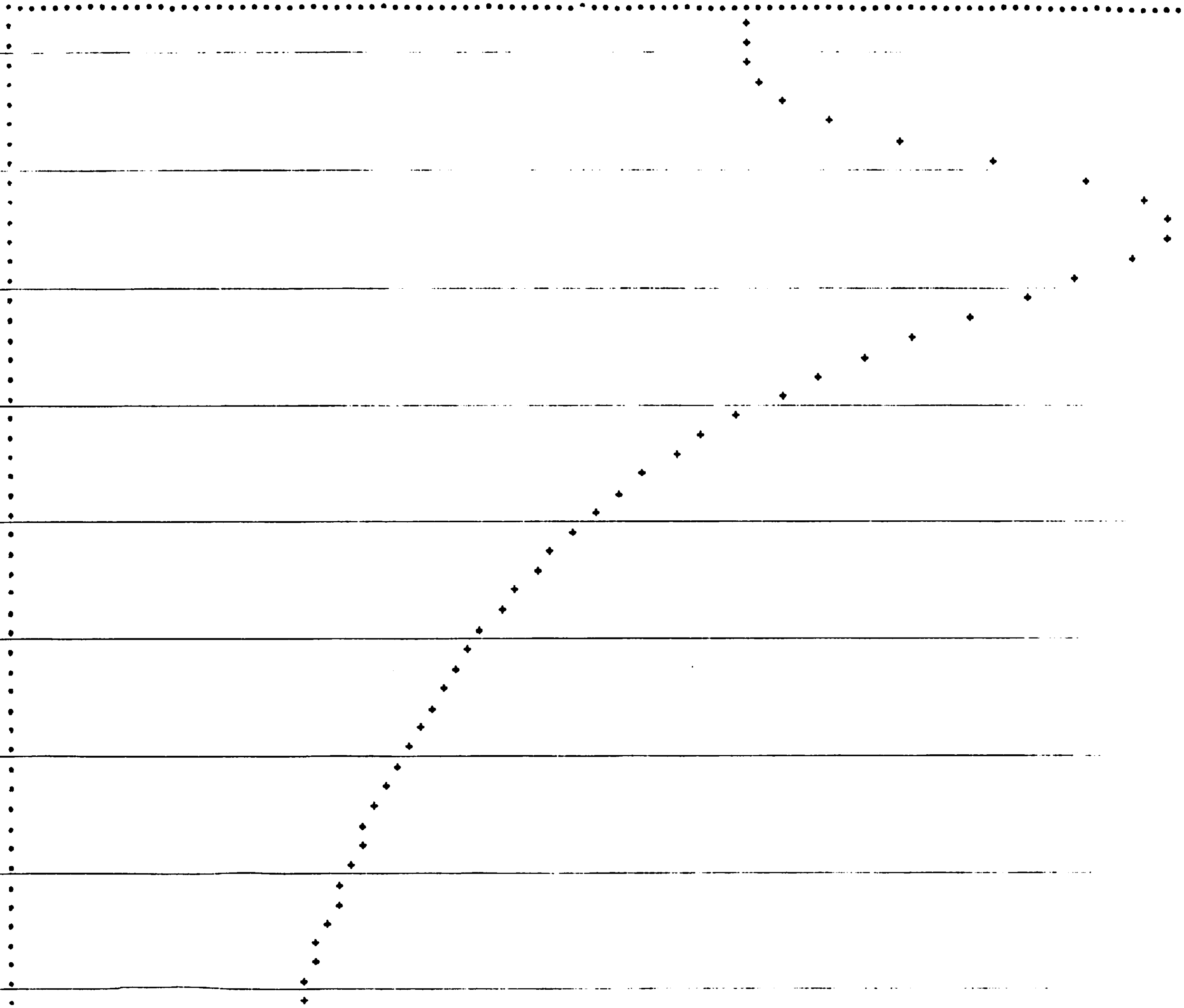
The required data cards for this problem may be seen on a subsequent page. The central processor time, for this problem was 9.86 seconds.

PROBLEM #1 – DATA CARDS:

```
    5    6         1000.0        5.0E+05                       1.0E-05    1    1
    5                            18        10000.0
E   1      2  1                               10.0
R   2      2  3              1000.0
L   3      3  4                0.01
C   4      4  1             2.0E-08
L   5      4  5                0.01
R   6      5  1              1000.0
    4    1    0
    1    1    0
    1    1    1
    6    0    0
    6    0    1
```

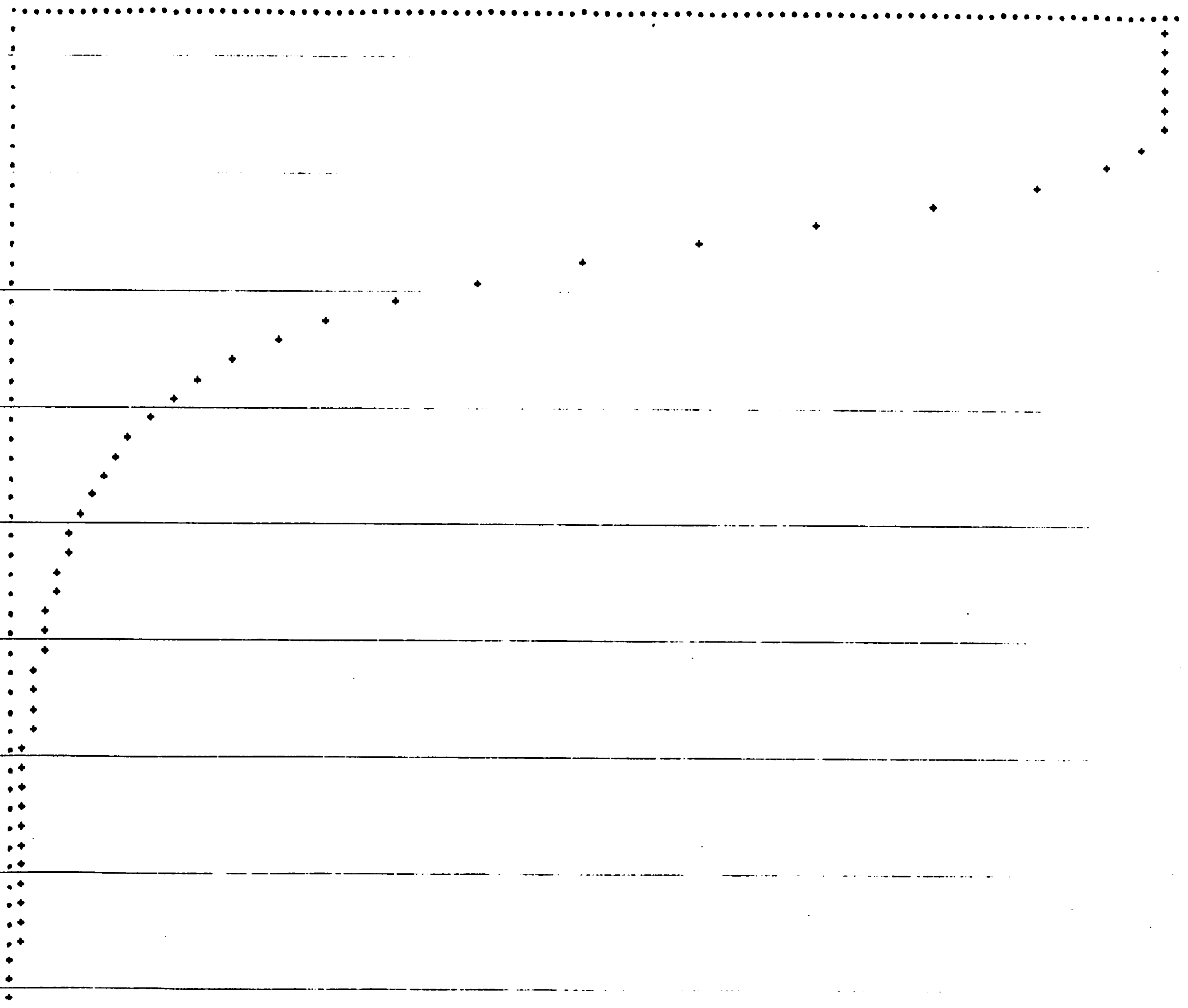
```
                           BRANCH  1      CURRENT        MAGNITUDE PLOT

FREQUENCY        VALUE
 .1000E+04      .5000E-02
 .1100E+05      .5001E-02
 .2100E+05      .5019E-02
 .3100E+05      .5089E-02
 .4100E+05      .5263E-02
 .5100E+05      .5587E-02
 .6100E+05      .6078E-02
 .7100E+05      .6645E-02
 .8100E+05      .7284E-02
 .9100E+05      .7725E-02
 .1010E+06      .7912E-02
 .1110E+06      .7848E-02
 .1210E+06      .7605E-02
 .1310E+06      .7265E-02
 .1410E+06      .6888E-02
 .1510E+06      .6511E-02
 .1610E+06      .6152E-02
 .1710E+06      .5818E-02
 .1810E+06      .5511E-02
 .1910E+06      .5231E-02
 .2010E+06      .4975E-02
 .2110E+06      .4742E-02
 .2210E+06      .4529E-02
 .2310E+06      .4334E-02
 .2410E+06      .4154E-02
 .2510E+06      .3989E-02
 .2610E+06      .3836E-02
 .2710E+06      .3694E-02
 .2810E+06      .3562E-02
 .2910E+06      .3440E-02
 .3010E+06      .3325E-02
 .3110E+06      .3218E-02
 .3210E+06      .3116E-02
 .3310E+06      .3023E-02
 .3410E+06      .2934E-02
 .3510E+06      .2851E-02
 .3610E+06      .2771E-02
 .3710E+06      .2697E-02
 .3810E+06      .2626E-02
 .3910E+06      .2559E-02
 .4010E+06      .2495E-02
 .4110E+06      .2434E-02
 .4210E+06      .2376E-02
 .4310E+06      .2321E-02
 .4410E+06      .2268E-02
 .4510E+06      .2218E-02
 .4610E+06      .2170E-02
 .4710E+06      .2124E-02
 .4810E+06      .2080E-02
 .4910E+06      .2037E-02
 .5010E+06      .1997E-02
```



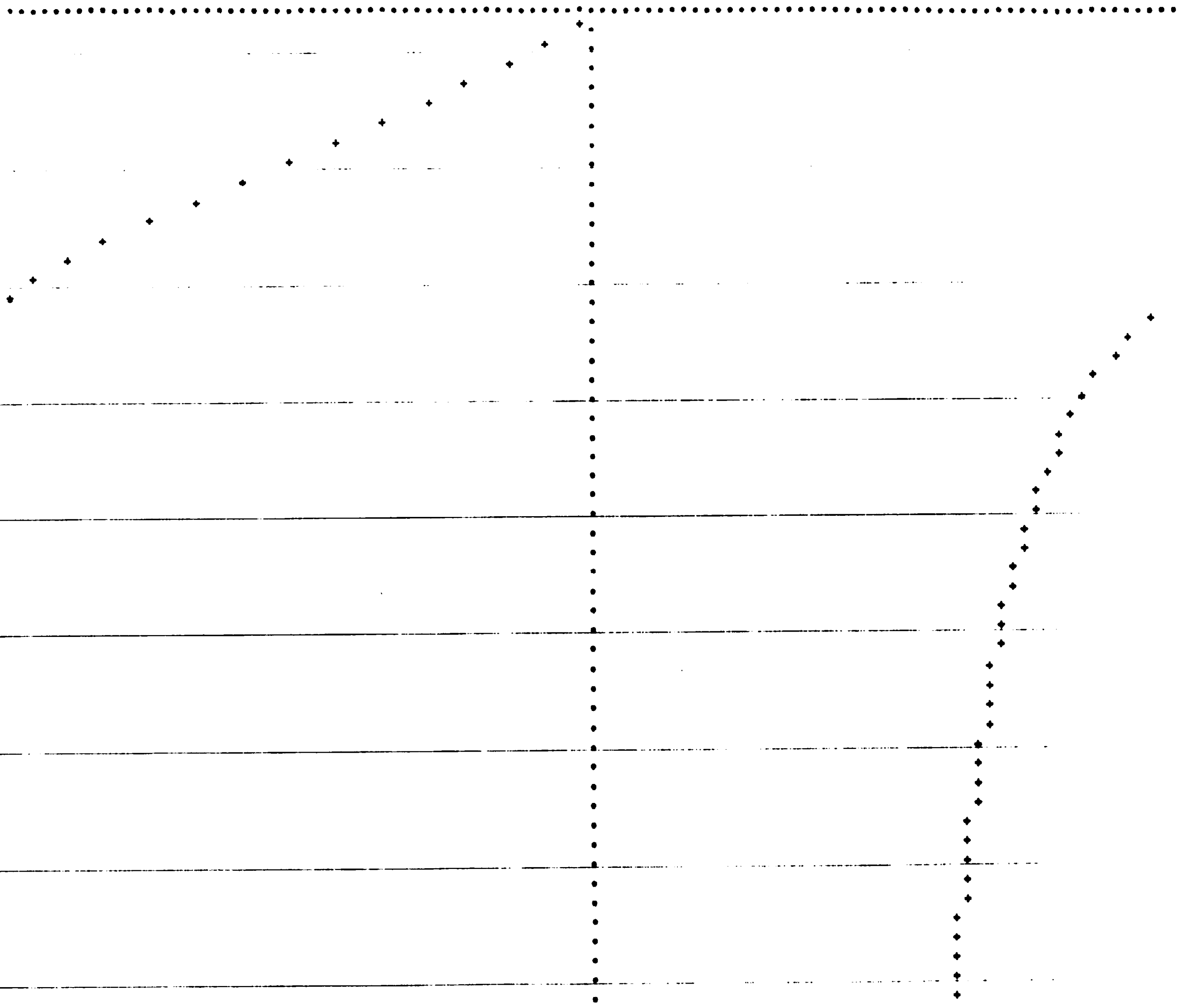
BRANCH 6 VOLTAGE MAGNITUDE PLOT

```
FREQUENCY     VALUE
 .1000E+04   .5000E+01
 .1100E+05   .5000E+01
 .2100E+05   .5000E+01
 .3100E+05   .4998E+01
 .4100E+05   .4988E+01
 .5100E+05   .4957E+01
 .6100E+05   .4876E+01
 .7100E+05   .4708E+01
 .8100E+05   .4415E+01
 .9100E+05   .3993E+01
 .1010E+06   .3482E+01
 .1110E+06   .2951E+01
 .1210E+06   .2458E+01
 .1310E+06   .2032E+01
 .1410E+06   .1680E+01
 .1510E+06   .1395E+01
 .1610E+06   .1165E+01
 .1710E+06   .9805E+00
 .1810E+06   .8315E+00
 .1910E+06   .7103E+00
 .2010E+06   .6111E+00
 .2110E+06   .5293E+00
 .2210E+06   .4613E+00
 .2310E+06   .4043E+00
 .2410E+06   .3563E+00
 .2510E+06   .3156E+00
 .2610E+06   .2808E+00
 .2710E+06   .2509E+00
 .2810E+06   .2251E+00
 .2910E+06   .2027E+00
 .3010E+06   .1832E+00
 .3110E+06   .1661E+00
 .3210E+06   .1511E+00
 .3310E+06   .1378E+00
 .3410E+06   .1261E+00
 .3510E+06   .1156E+00
 .3610E+06   .1063E+00
 .3710E+06   .9790E-01
 .3810E+06   .9601E-01
 .3910E+06   .8892E-01
 .4010E+06   .8226E-01
 .4110E+06   .7630E-01
 .4210E+06   .7091E-01
 .4310E+06   .6601E-01
 .4410E+06   .6155E-01
 .4510E+06   .5875E-01
 .4610E+06   .5522E-01
 .4710E+06   .5151E-01
 .4810E+06   .4838E-01
 .4910E+06   .4539E-01
 .5010E+06   .4220E-01
```

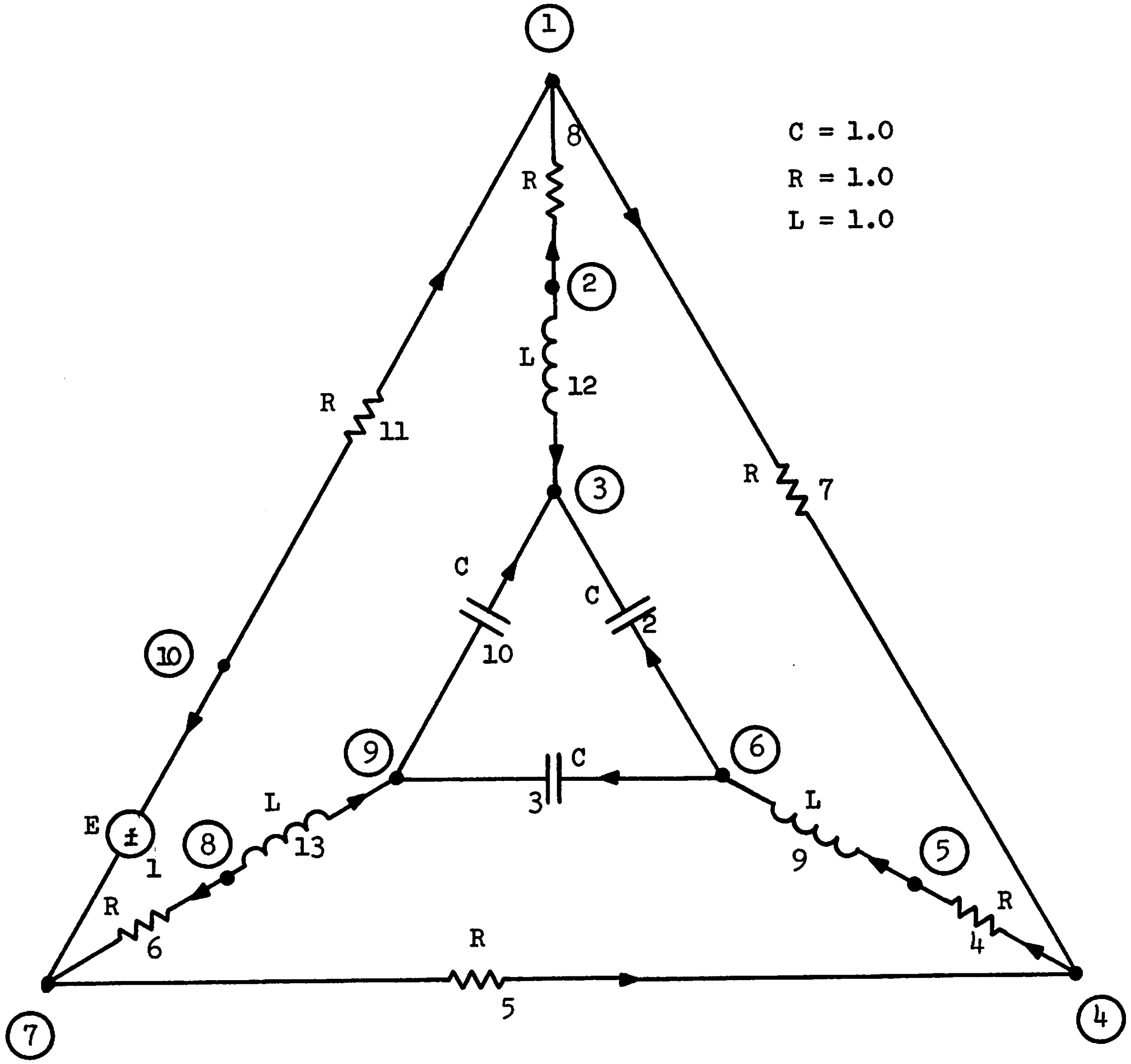


```
                  BRANCH  6      VOLTAGE      PHASE PLOT

 FREQUENCY      VALUE
  .1000E+04   -.1175E+01
  .1100E+05   -.1263E+02
  .2100E+05   -.2425E+02
  .3100E+05   -.3615E+02
  .4100E+05   -.4853E+02
  .5100E+05   -.6160E+02
  .6100E+05   -.7555E+02
  .7100E+05   -.9044E+02
  .8100E+05   -.1060E+03
  .9100E+05   -.1216E+03
  .1010E+06   -.1364E+03
  .1110E+06   -.1498E+03
  .1210E+06   -.1614E+03
  .1310E+06   -.1713E+03
  .1410E+06   -.1797E+03
  .1510E+06    .1732E+03
  .1610E+06    .1672E+03
  .1710E+06    .1619E+03
  .1810E+06    .1574E+03
  .1910E+06    .1534E+03
  .2010E+06    .1499E+03
  .2110E+06    .1468E+03
  .2210E+06    .1440E+03
  .2310E+06    .1415E+03
  .2410E+06    .1392E+03
  .2510E+06    .1371E+03
  .2610E+06    .1351E+03
  .2710E+06    .1334E+03
  .2810E+06    .1318E+03
  .2910E+06    .1303E+03
  .3010E+06    .1289E+03
  .3110E+06    .1276E+03
  .3210E+06    .1263E+03
  .3310E+06    .1252E+03
  .3410E+06    .1241E+03
  .3510E+06    .1231E+03
  .3610E+06    .1222E+03
  .3710E+06    .1213E+03
  .3810E+06    .1202E+03
  .3910E+06    .1194E+03
  .4010E+06    .1187E+03
  .4110E+06    .1180E+03
  .4210E+06    .1173E+03
  .4310E+06    .1166E+03
  .4410E+06    .1160E+03
  .4510E+06    .1150E+03
  .4610E+06    .1146E+03
  .4710E+06    .1140E+03
  .4810E+06    .1136E+03
  .4910E+06    .1131E+03
  .5010E+06    .1133E+03
```



Sample Problem #2:



|E| = 1

∢ E = 0.0°

NN = 10

NB = 13

Specifications:

- Initial frequency = 0.0 (WSTART = 0.0)
- Final frequency = 2.0 (WEND = 2.0)
- Frequency additive step = 0.1 (FREQADD = 0.1)
- Desire error to fall below 10^{-5} or a maximum of 39 iterative minimization steps at each frequency point. (EPS = 10^{-5}, NITT = 39)
- Do not desire tree and $\underset{\wedge}{F}$ to be outputed (NCONT = 0)
- Desire internal automatic current scaling (NSCALE $\neq$ 1, SCALE is arbitrary)
- Do not desire an optimal tree to be selected at each frequency (NTREE $\neq$ 1)
- Do not desire testing for ideal resonance (IDRES = 1)
- Desire the following graphical outputs

$$\left.\begin{array}{l}\text{Branch 3 current magnitude}\\ \text{Branch 5 voltage phase}\\ \text{Branch 8 voltage magnitude}\\ \text{Branch 11 current magnitude}\\ \text{Branch 13 current phase}\end{array}\right\}\ \text{(NGRAPH = 5)}$$

The required data cards, for this problem, may be seen on a subsequent page. The central processor time for this problem was 10.37 seconds.

Comments on sample problem #2:

Note that when $\omega = 0$, only the outer resistor loop can carry a current (since all the C's are open circuits). Thus, the current in branch 11 is 1/3 amp, and the voltage across branch 8 is zero. The results of CANDOFD agree with the above. As $\omega \rightarrow \infty$, the current in branch 11 will again tend toward 1/3, and the voltage across branch 8 will again tend toward zero (since the L's now become open circuits).

Comparison of different solution schemes for sample problem #2:

* implies that the indicated scheme has been implemented

CP indicates the central processor time, in seconds

AUTO indicates automatic, internal current scaling

TREE indicates that an optimal tree is to be selected at each frequency point

OUT indicates that tree and $\hat{F}$ outputs are desired

RES indicates that a test for resonances is desired

Scheme	AUTO	TREE	RES	OUT	CP
1	*	*	*	*	13.54
2	*	*	*		10.98
3	*	*			10.91
4	*				10.37
5	SCALE = 1.0	*			10.64
6	SCALE = 1.0				10.30

It should be noted from a comparison of the 1st and 2nd schemes, that the CP time for outputing the tree and $\hat{F}$ information is substantial, and hence should not be used unless it is specifically desired. Even though schemes 4 has a CP time slightly below that of scheme 3, its results are significantly less accurate.

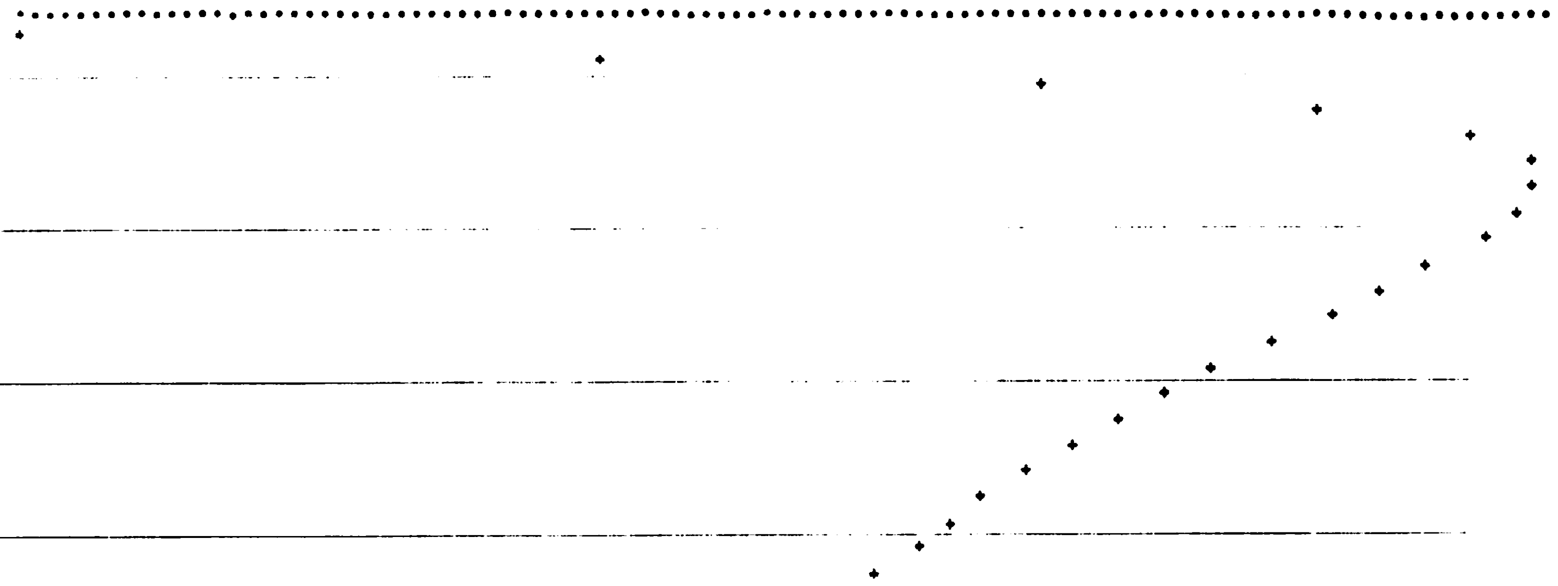
In general, scheme 3 is the recommended procedure when one does not have a clear idea of what the network signals will be.

PROBLEM #2: DATA CARDS

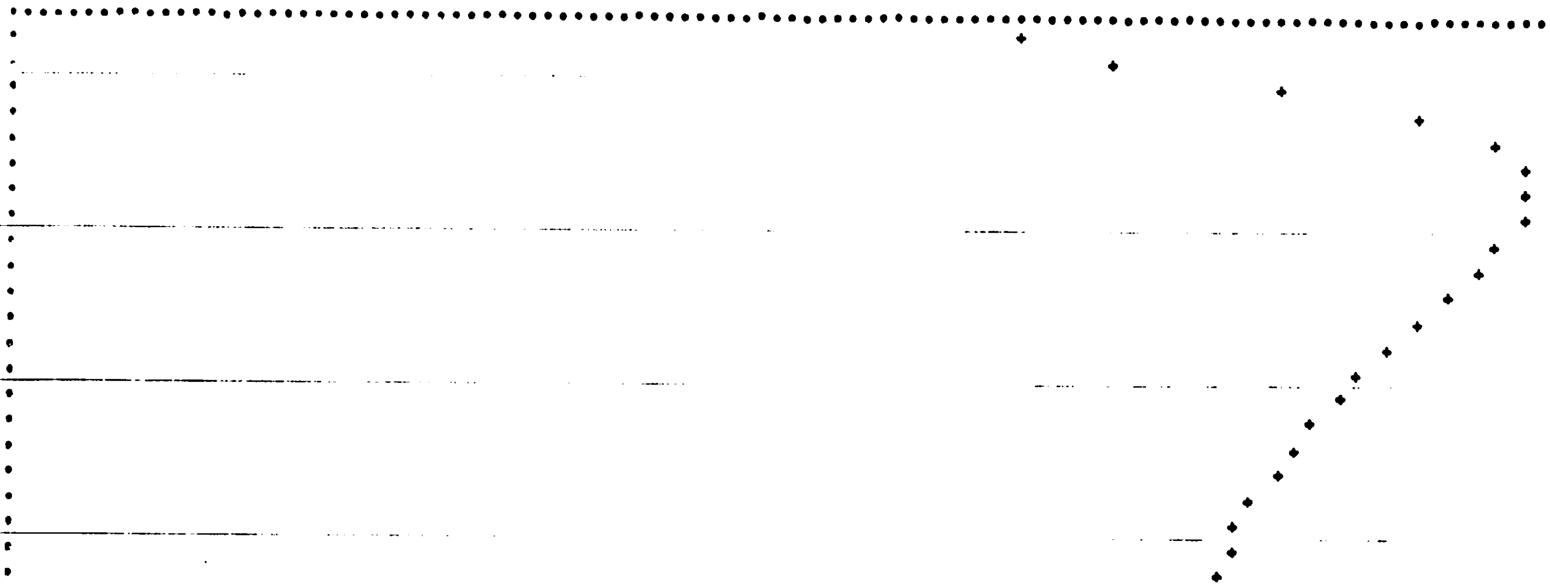
```
   10   13            0.0            2.0                       1.0E-05
    5                            39            0.1         1
E  1      10  7                                0.1
C  2       6  3                  1.0
C  3       6  9                  1.0
R  4       4  5                  1.0
R  5       7  4                  1.0
R  6       8  7                  1.0
R  7       1  4                  1.0
R  8       2  1                  1.0
L  9       5  6                  1.0
C 10       9  3                  1.0
R 11      10  1                  1.0
L 12       2  3                  1.0
L 13       8  9                  1.0
    3    1    0
    5    0    1
    8    0    0
   11    1    0
   13    1    1
```

```
                    BRANCH  B      VOLTAGE      MAGNITUDE PLOT

FREQUENCY        VALUE
0.               0.
 .1000E+00        .9531E-01
 .2000E+00        .1682E+00
 .3000E+00        .2135E+00
 .4000E+00        .2378E+00
 .5000E+00        .2486E+00
 .6000E+00        .2499E+00
 .7000E+00        .2465E+00
 .8000E+00        .2403E+00
 .9000E+00        .2323E+00
 .1000E+01        .2236E+00
 .1100E+01        .2155E+00
 .1200E+01        .2055E+00
 .1300E+01        .1968E+00
 .1400E+01        .1897E+00
 .1500E+01        .1804E+00
 .1600E+01        .1726E+00
 .1700E+01        .1669E+00
 .1800E+01        .1597E+00
 .1900E+01        .1527E+00
 .2000E+01        .1498E+00
 .2100E+01        .1423E+00
```

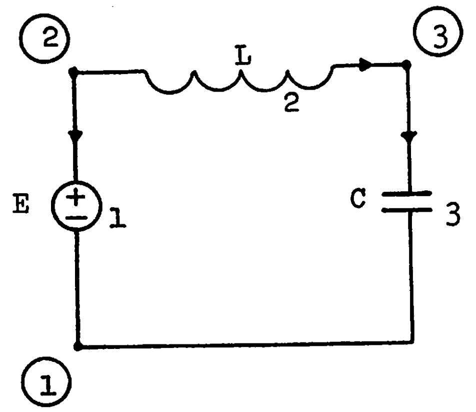


```
             BRANCH 11     CURRENT     MAGNITUDE PLOT
FREQUENCY      VALUE
0.            .3333E+00
 .1000E+00    .3623E+00
 .2000E+00    .4171E+00
 .3000E+00    .4610E+00
 .4000E+00    .4866E+00
 .5000E+00    .4971E+00
 .6000E+00    .4998E+00
 .7000E+00    .4962E+00
 .8000E+00    .4893E+00
 .9000E+00    .4807E+00
 .1000E+01    .4714E+00
 .1100E+01    .4615E+00
 .1200E+01    .4524E+00
 .1300E+01    .4443E+00
 .1400E+01    .4366E+00
 .1500E+01    .4284E+00
 .1600E+01    .4218E+00
 .1700E+01    .4165E+00
 .1800E+01    .4097E+00
 .1900E+01    .4044E+00
 .2000E+01    .4025E+00
 .2100E+01    .3953E+00
```



Sample Problem #3:

Ideal L-C filter section:



L = 1.0

C = 1.0

NN = 3

NB = 3

$|E| = 1.0$, $\measuredangle E = 0.0$

Specifications:

- Initial frequency = 0.0 (WSTART = 0.0)
- Final frequency = 2.5 (WEND = 2.5)
- Frequency additive factor = 0.1 (FREQADD = 0.1)
- Desire error to fall below 10^{-5} (EPS = 10^{-5})
- Desire tree and $\underset{\wedge}{F}$ to be outputed (NCONT = 1)
- Desire internal automatic current scaling (SCALE is arbitrary, NSCALE $\neq$ 1)
- Desire an optimal tree to be selected at each frequency point (NTREE = 1)
- Desire the following graphical outputs

  Branch 1 voltage magnitude

  Branch 2 current magnitude

  Branch 2 current phase

  Branch 3 voltage magnitude

  Branch 3 voltage phase

  (NGRAPH = 5)

The required data cards for this problem may be seen on a subsequent page. The central processor time for this problem was 2.61 seconds.

Comments on Sample Problem #3:

Note the XXX's on the graphical outputs indicating an ideal resonance at $\omega = 1.0$.

PROBLEM #3: DATA CARDS:

```
    3    3            0.0            2.5                       1.0E-05         1
    5                                          0.1
E  1       2  1                                1.0
L  2       2  3                 1.0
C  3       3  1                 1.0
    1    0    0
    2    1    0
    2    1    1
    3    0    0
    3    0    1
```

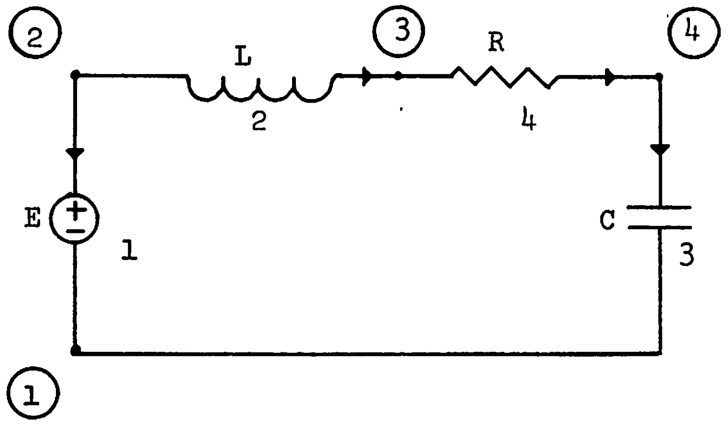
```
                              BRANCH  3     VOLTAGE      MAGNITUDE PLOT

FREQUENCY      VALUE        ....................................................................................................
0.             .1000E+01    .         +
 .1000E+00     .1010E+01    .         +
 .2000E+00     .1042E+01    .         +
 .3000E+00     .1099E+01    .          +
 .4000E+00     .1190E+01    .           +
 .5000E+00     .1333E+01    .            +
 .6000E+00     .1562E+01    .               +
 .7000E+00     .1961E+01    .                   +
 .8000E+00     .2778E+01    .                           +
 .9000E+00     .5263E+01    .                                                    +
 .1050E+01     .9756E+01    .   XXXXXXXXXXX                                                                                   +
 .1100E+01     .4762E+01    .                                               +
 .1200E+01     .2273E+01    .                      +
 .1300E+01     .1449E+01    .             +
 .1400E+01     .1042E+01    .         +
 .1500E+01     .8000E+00    .       +
 .1600E+01     .6410E+00    .     +
 .1700E+01     .5291E+00    .    +
 .1800E+01     .4464E+00    .   +
 .1900E+01     .3831E+00    .  +
 .2000E+01     .3333E+00    .  +
 .2100E+01     .2933E+00    .  +
 .2200E+01     .2604E+00    . +
 .2300E+01     .2331E+00    . +
 .2400E+01     .2101E+00    . +
 .2500E+01     .1905E+00    .+
 .2600E+01     .1736E+00    .+
                                BRANCH  3      VOLTAGE       PHASE PLOT

   FREQUENCY      VALUE       .....................................................................................................
   0.           0.
    .1000E+00   0.                                                                 +
    .2000E+00   0.                                                                 +
    .3000E+00   0.                                                                 +
    .4000E+00   0.                                                                 +
    .5000E+00   0.                                                                 +
    .6000E+00   0.                                                                 +
    .7000E+00   0.                                                                 +
    .8000E+00   0.                                                                 +
    .9000E+00   0.                                                                 +
    .1050E+01   -.1800E+03   +   XXXXXXXXXXX
    .1100E+01   -.1800E+03   +
    .1200E+01   -.1800E+03   +
    .1300E+01   -.1800E+03   +
    .1400E+01   -.1800E+03   +
    .1500E+01   -.1800E+03   +
    .1600E+01   -.1800E+03   +
    .1700E+01   -.1800E+03   +
    .1800E+01   -.1800E+03   +
    .1900E+01   -.1800E+03   +
    .2000E+01   -.1800E+03   +
    .2100E+01   -.1800E+03   +
    .2200E+01   -.1800E+03   +
    .2300E+01   -.1800E+03   +
    .2400E+01   -.1800E+03   +
    .2500E+01   -.1800E+03   +
    .2600E+01   -.1800E+03   +
```

Sample Problem #4:

Lossy L-C filter section:



R = 0.1

L = 1.0

C = 1.0

NN = 4

NB = 4

$|E| = 1.0, \measuredangle E = 0.0$

Specifications

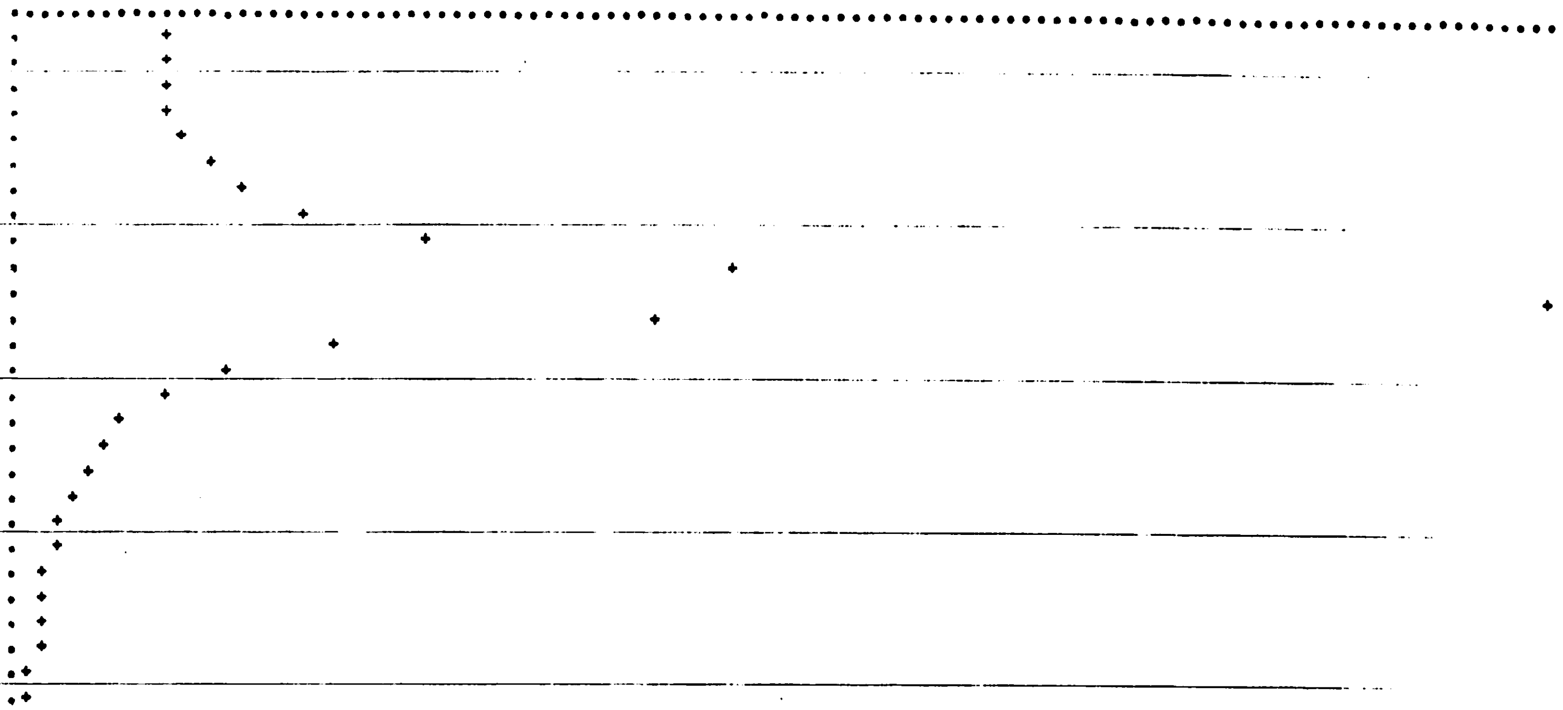
- Initial frequency = 0.0 (WSTART = 0.0)
- Final frequency = 2.5 (WEND = 2.5)
- Frequency additive factor = 0.1 (FREQADD = 0.1)
- Desire error to fall below 10^{-5} (EPS = 10^{-5})
- Do not desire tree and $\underset{\wedge}{F}$ to be outputed (NCONT = 0)
- Desire internal automatic current scaling (SCALE is arbitrary, NSCALE ≠ 1)
- Desire an optimal tree to be selected at each frequency point (NTREE = 1)
- Desire the following graphical outputs.

SAMPLE PROBLEM #4: DATA CARDS

```
    4    4            0.0            2.5                       1.0E-05         1
    5                                          0.1
E   1      2  1                                1.0
L   2      2  3                 1.0
C   3      4  1                 1.0
R   4      3  4                 0.1
    1    0    0
    2    1    1
    2    1    0
    3    0    0
    3    0    1
```

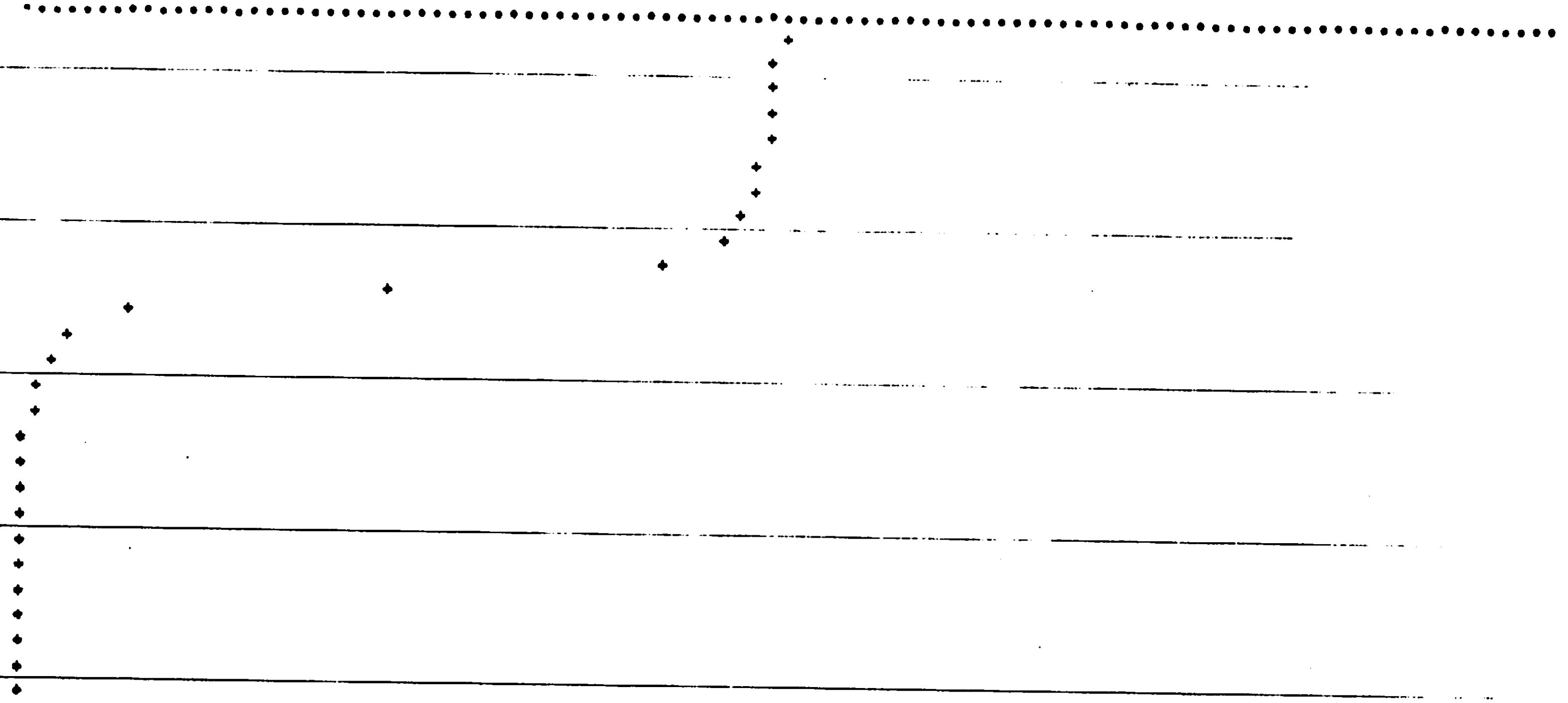
```
                              BRANCH  3     VOLTAGE      MAGNITUDE PLOT

FREQUENCY         VALUE
0.               .1000E+01
 .1000E+00       .1010E+01
 .2000E+00       .1041E+01
 .3000E+00       .1098E+01
 .4000E+00       .1189E+01
 .5000E+00       .1330E+01
 .6000E+00       .1556E+01
 .7000E+00       .1943E+01
 .8000E+00       .2712E+01
 .9000E+00       .4757E+01
 .1000E+01       .1000E+02
 .1100E+01       .4218E+01
 .1200E+01       .2193E+01
 .1300E+01       .1424E+01
 .1400E+01       .1031E+01
 .1500E+01       .7943E+00
 .1600E+01       .6377E+00
 .1700E+01       .5270E+00
 .1800E+01       .4450E+00
 .1900E+01       .3821E+00
 .2000E+01       .3326E+00
 .2100E+01       .2927E+00
 .2200E+01       .2600E+00
 .2300E+01       .2328E+00
 .2400E+01       .2098E+00
 .2500E+01       .1903E+00
 .2600E+01       .1734E+00
```

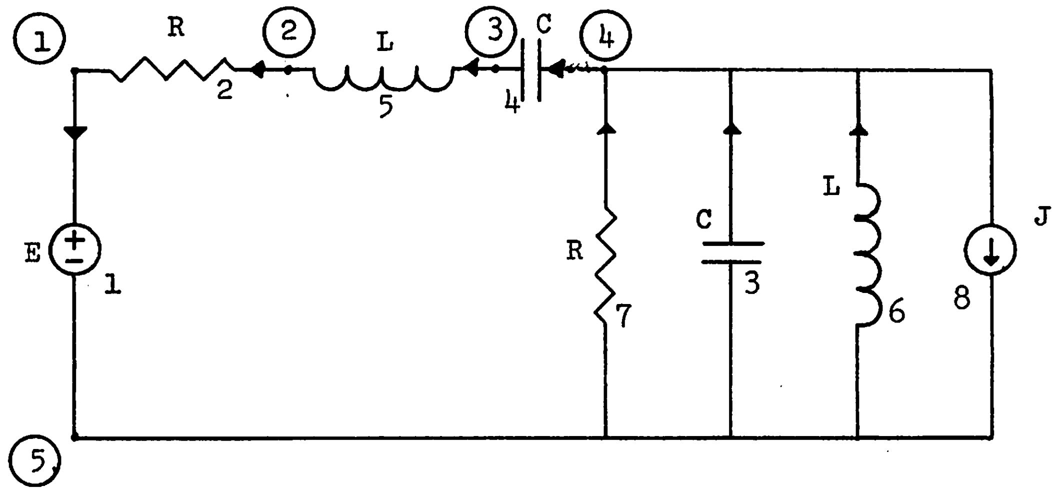


BRANCH 3 VOLTAGE PHASE PLOT

```
FREQUENCY      VALUE
0.             0.
 .1000E+00     -.5788E+00
 .2000F+00     -.1193F+01
 .3000F+00     -.1888E+01
 .4000F+00     -.2726E+01
 .5000E+00     -.3814E+01
 .6000E+00     -.5356E+01
 .7000F+00     -.7815E+01
 .8000E+00     -.1253E+02
 .9000E+00     -.2535E+02
 .1000E+01     -.9000E+02
 .1100E+01     -.1524E+03
 .1200E+01     -.1647E+03
 .1300E+01     -.1693E+03
 .1400E+01     -.1717E+03
 .1500E+01     -.1732E+03
 .1600E+01     -.1741E+03
 .1700E+01     -.1749E+03
 .1800E+01     -.1754E+03
 .1900E+01     -.1758E+03
 .2000E+01     -.1762E+03
 .2100E+01     -.1765E+03
 .2200E+01     -.1767E+03
 .2300E+01     -.1769E+03
 .2400E+01     -.1771E+03
 .2500E+01     -.1773E+03
 .2600E+01     -.1774E+03
```



Sample Problem #5:



$|E| = |J| = 10.0$ R = 1.0

$\measuredangle E = \measuredangle J = 0°$ L = 0.05

NN = 5, NB = 8 C = 0.05

Specifications:

- Initial frequency = 0.0 (WSTART = 0.0)
- Final frequency = 50.0 (WEND = 50.0)
- Frequency additive factor = 1.0 (FREQADD = 1.0)
- Desire error to fall below 10^{-4} or a maximum of 24 iterative minimization steps at each frequency point. (EPS = 10^{-4}, NITT = 24)
- Do not desire tree and $\hat{F}$ to be outputed (NCONT = 0)
- Desire internal automatic current scaling. (SCALE is arbitrary, NSCALE ≠ 1)
- Desire an optimal tree to be selected at each frequency point (NTREE = 1)

- Desire the following graphical outputs

  Branch 1 current magnitude
  Branch 2 voltage magnitude
  Branch 5 current magnitude
  Branch 5 current phase
  Branch 8 current magnitude

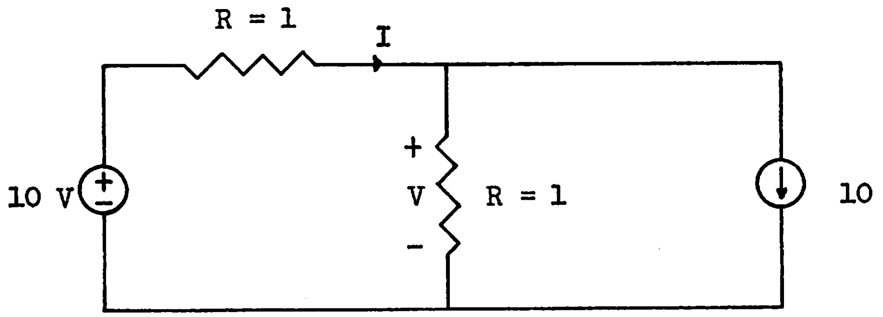
  (NGRAPH = 5)

- Desire internal testing for resonances (IDRES ≠ 1)

The required data cards for this problem may be seen on a subsequent page. The central processor time for this problem was 10.56 seconds.

Comments on sample problem #5:

At $\omega = 20.0$, the L-C series and parallel combinations act as short and open circuits respectfully. Thus, our network reduces to



The solution to this network is

$V = 0$ volts

$I = 10$ amps

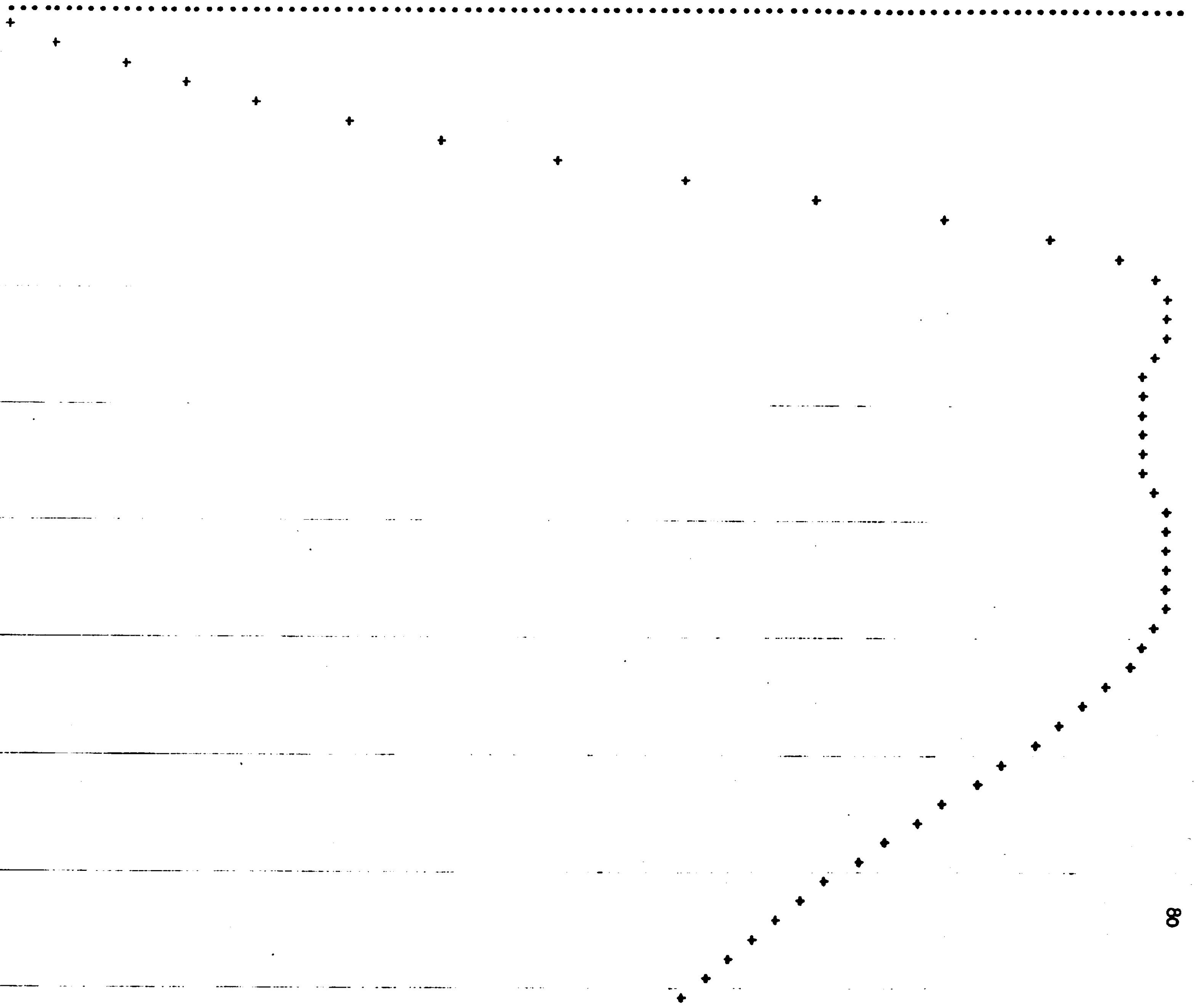
This is borne out by CANDOFD.

DATA CARDS: SAMPLE PROBLEM # 5:

```
    5    8            0.0           50.0                        0.0001         1
    5                            24            1.0
E  1       1  5                               10.0
R  2       2  1                 1.0
J  8       4  5                               10.0
R  7       5  4                 1.0
L  5       3  2                0.05
L  6       5  4                0.05
C  3       5  4                0.05
C  4       4  3                0.05
    1    1    0
    2    0    0
    5    1    0
    5    1    1
    8    1    0
```

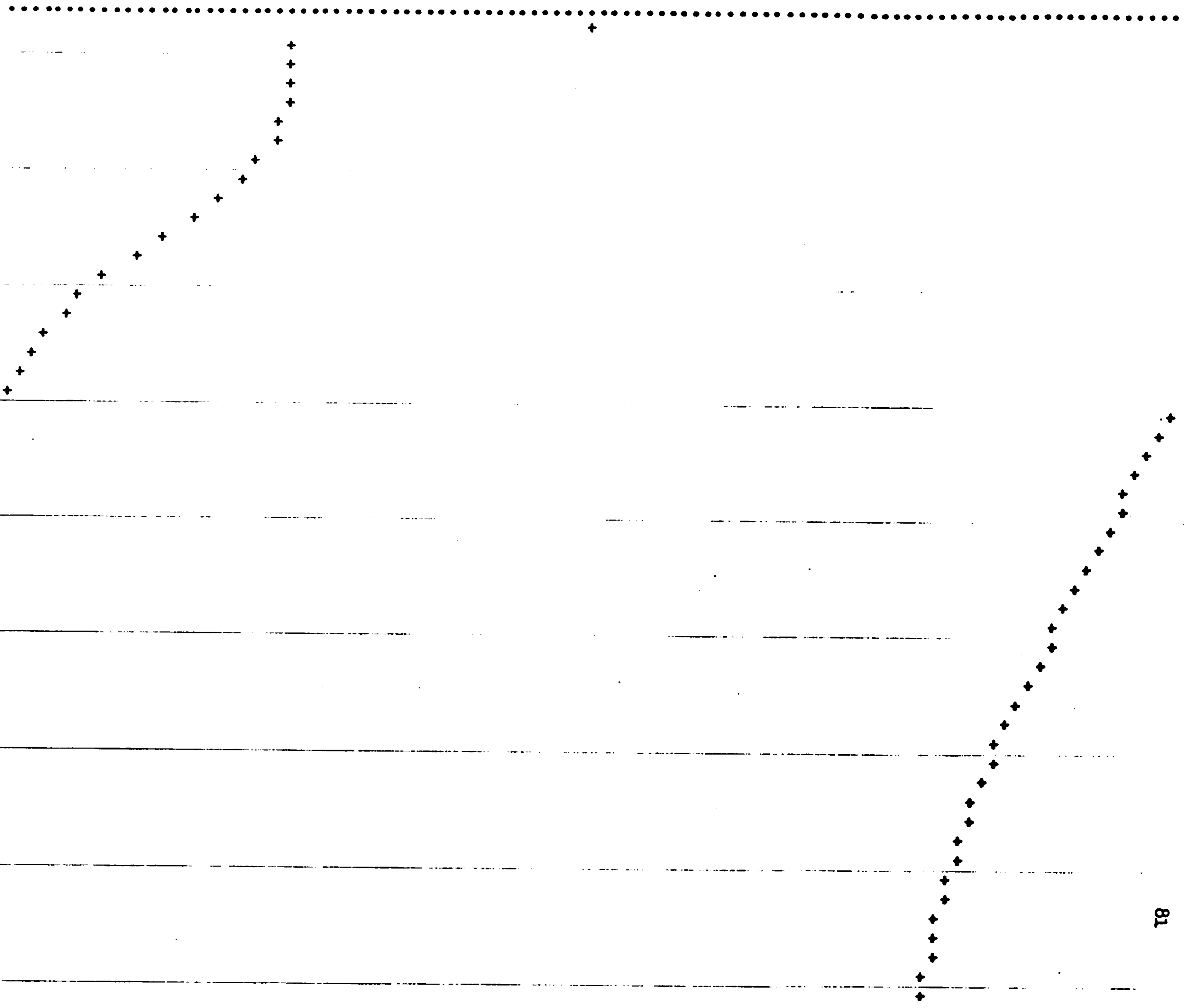
BRANCH   5   CURRENT   MAGNITUDE PLOT

```
FREQUENCY        VALUE
0.               0.
 .1000E+01        .5038E+00
 .2000E+01        .1030E+01
 .3000E+01        .1603E+01
 .4000E+01        .2248E+01
 .5000E+01        .2992E+01
 .6000E+01        .3859E+01
 .7000E+01        .4862E+01
 .8000E+01        .5989E+01
 .9000E+01        .7177E+01
 .1000E+02        .8305E+01
 .1100E+02        .9228E+01
 .1200E+02        .9851E+01
 .1300E+02        .1018E+02
 .1400E+02        .1029E+02
 .1500E+02        .1027E+02
 .1600E+02        .1020E+02
 .1700E+02        .1012E+02
 .1800E+02        .1005E+02
 .1900E+02        .1001E+02
 .2000E+02        .1000E+02
 .2100E+02        .1001E+02
 .2200E+02        .1004E+02
 .2300E+02        .1008E+02
 .2400E+02        .1014E+02
 .2500E+02        .1020E+02
 .2600E+02        .1025E+02
 .2700E+02        .1028E+02
 .2800E+02        .1029E+02
 .2900E+02        .1028E+02
 .3000E+02        .1023E+02
 .3100E+02        .1016E+02
 .3200E+02        .1005E+02
 .3300E+02        .9905E+01
 .3400E+02        .9733E+01
 .3500E+02        .9533E+01
 .3600E+02        .9312E+01
 .3700E+02        .9074E+01
 .3800E+02        .8824E+01
 .3900E+02        .8567E+01
 .4000E+02        .8305E+01
 .4100E+02        .8044E+01
 .4200E+02        .7785E+01
 .4300E+02        .7531E+01
 .4400E+02        .7284E+01
 .4500E+02        .7045E+01
 .4600E+02        .6813E+01
 .4700E+02        .6594E+01
 .4800E+02        .6383E+01
 .4900E+02        .6181E+01
 .5000E+02        .5989E+01
```



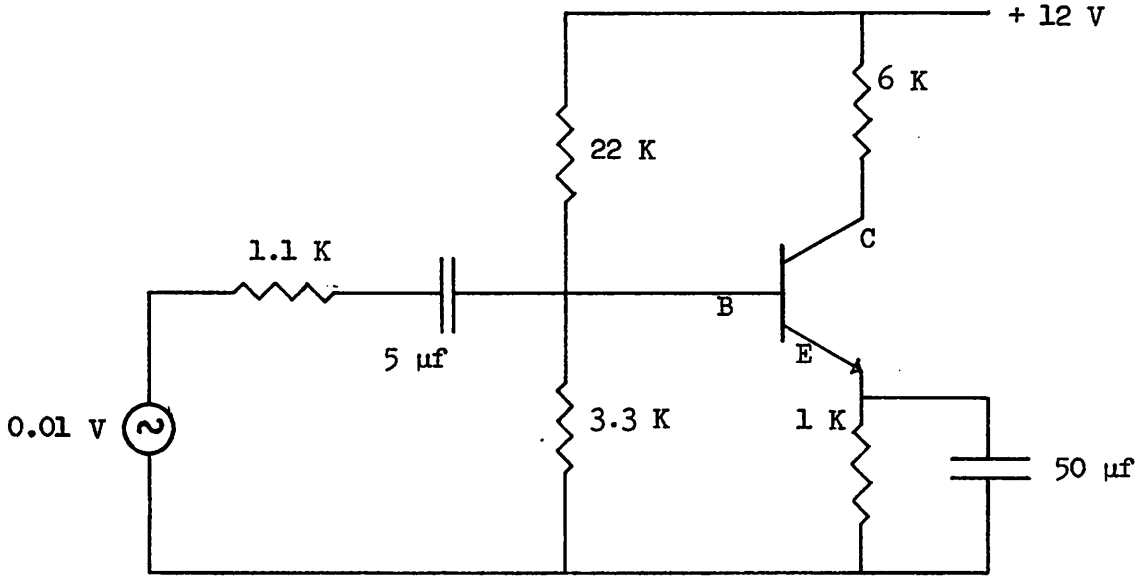
BRANCH 5 CURRENT PHASE PLOT

```
FREQUENCY        VALUE
0.               0.
 .1000E+01      -.9003E+02
 .2000E+01      -.9023E+02
 .3000E+01      -.9079E+02
 .4000E+01      -.9191E+02
 .5000E+01      -.9380E+02
 .6000E+01      -.9672E+02
 .7000E+01      -.1009E+03
 .8000E+01      -.1066E+03
 .9000E+01      -.1137E+03
 .1000E+02      -.1221E+03
 .1100E+02      -.1312E+03
 .1200E+02      -.1401E+03
 .1300E+02      -.1482E+03
 .1400E+02      -.1553E+03
 .1500E+02      -.1612E+03
 .1600E+02      -.1661E+03
 .1700E+02      -.1702E+03
 .1800E+02      -.1738E+03
 .1900E+02      -.1770E+03
 .2000E+02       .1800E+03
 .2100E+02       .1772E+03
 .2200E+02       .1744E+03
 .2300E+02       .1717E+03
 .2400E+02       .1689E+03
 .2500E+02       .1661E+03
 .2600E+02       .1632E+03
 .2700E+02       .1601E+03
 .2800E+02       .1570E+03
 .2900E+02       .1539E+03
 .3000E+02       .1507E+03
 .3100E+02       .1475E+03
 .3200E+02       .1443E+03
 .3300E+02       .1411E+03
 .3400E+02       .1380E+03
 .3500E+02       .1350E+03
 .3600E+02       .1322E+03
 .3700E+02       .1294E+03
 .3800E+02       .1268E+03
 .3900E+02       .1244E+03
 .4000E+02       .1221E+03
 .4100E+02       .1200E+03
 .4200E+02       .1180E+03
 .4300E+02       .1161E+03
 .4400E+02       .1144E+03
 .4500E+02       .1128E+03
 .4600E+02       .1114E+03
 .4700E+02       .1100E+03
 .4800E+02       .1088E+03
 .4900E+02       .1076E+03
 .5000E+02       .1065E+03
```



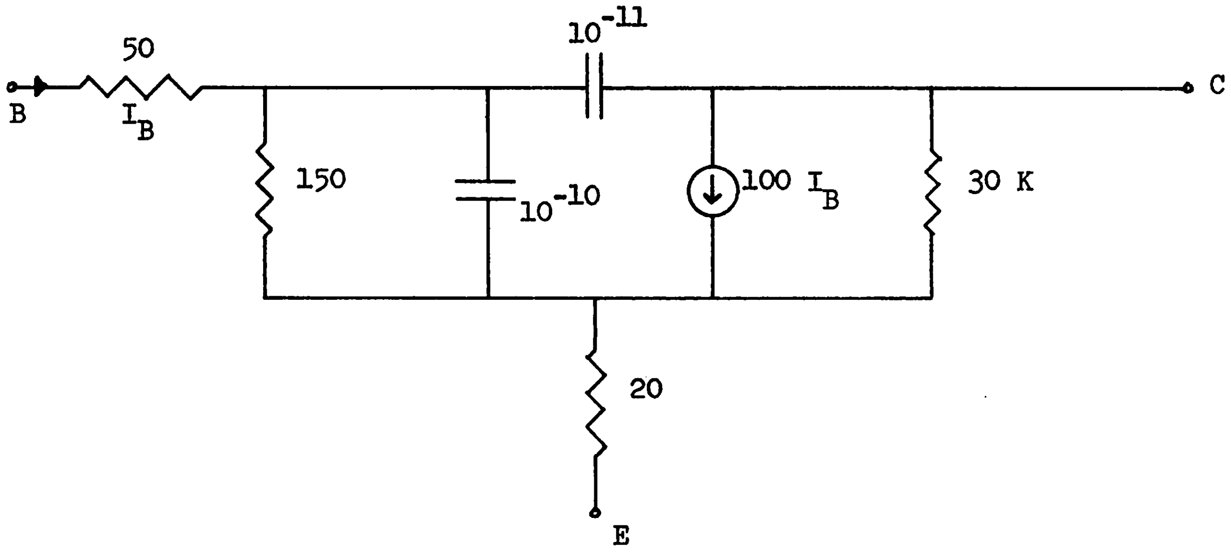
Sample Problem #6:

One stage amplifier:

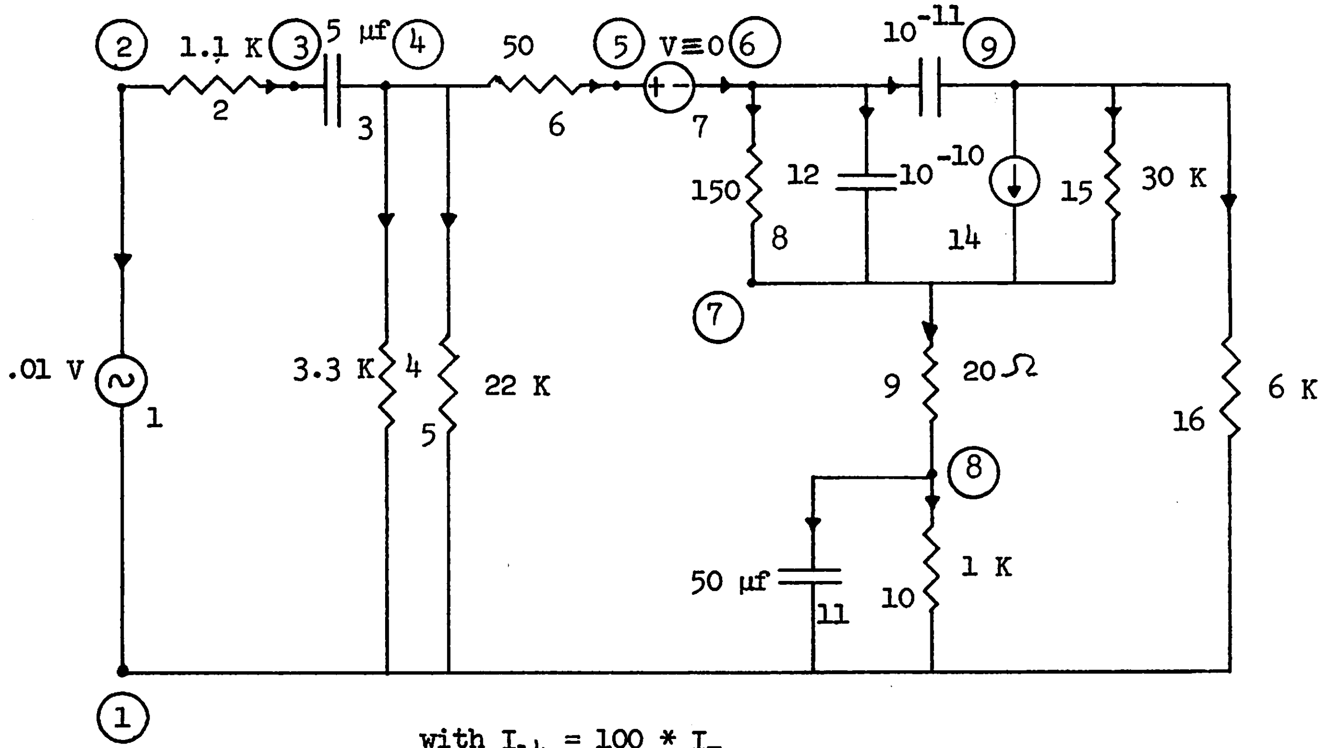


where µf indicates microfarads and K indicates ohms × 10^3.

The transistor is represented by the following small signal model:



With the above transistor model and appropriate controlled source modeling, the small signal equivalent of our network is;



with $I_{14} = 100 * I_7$

NN = 9, NB = 16

Specifications:

- Initial frequency = 1.0 (WSTART = 1.0)
- Final frequency = 10^9 (WEND = 10^9)
- Frequency multiplicative factor = 2.0 (FACTOR = 2.0)
- Desire error to fall below 10^{-12} or a maximum of 50 iterative minimization steps at each frequency point (EPS = 10^{-12}, NITT = 50)
- Do not desire tree and $\underset{\sim}{F}$ information outputed (NCONT ≠ 1)
- Desire internal, automatic current scaling (SCALE is arbitrary and NSCALE ≠ 1)
- Desire the selection of an optimal tree at each frequency point. (NTREE = 1)
- Do not desire testing for ideal resonances (IDRES = 1)

- Desire the following graphical outputs

$$\left.\begin{array}{l}\text{Branch 1 current magnitude}\\ \text{Branch 7 current magnitude}\\ \text{Branch 14 current magnitude}\\ \text{Branch 16 current magnitude}\end{array}\right\}\ (\text{NGRAPH} = 4)$$

The required data cards for this problem may be seen on a subsequent page. The central processor time, for this problem, was 69 seconds, i.e., approximately 2 seconds per frequency point. (Note that this is, essentially, an exact analysis.)

Comments on sample problem #6:

The mid-band voltage gain, given by CANDOFD, is

$$G_V = \frac{R_{16} \times I_{16}}{V_1} = \frac{6 \times 10^3 \times 0.22 \times 10^{-3}}{0.01} \cong 132$$

where $R_{16} = 6$ K.

The mid-band power gain, given by CANDOFD, is

$$G_P = \frac{R_{16} \times I_{16}^{\,2}}{} = G_V \times \frac{I_{16}}{I_1}$$

$$= 132 \times \frac{0.22 \times 10^{-3}}{0.44 \times 10^{-5}}$$

$$\cong 6.6 \times 10^3$$

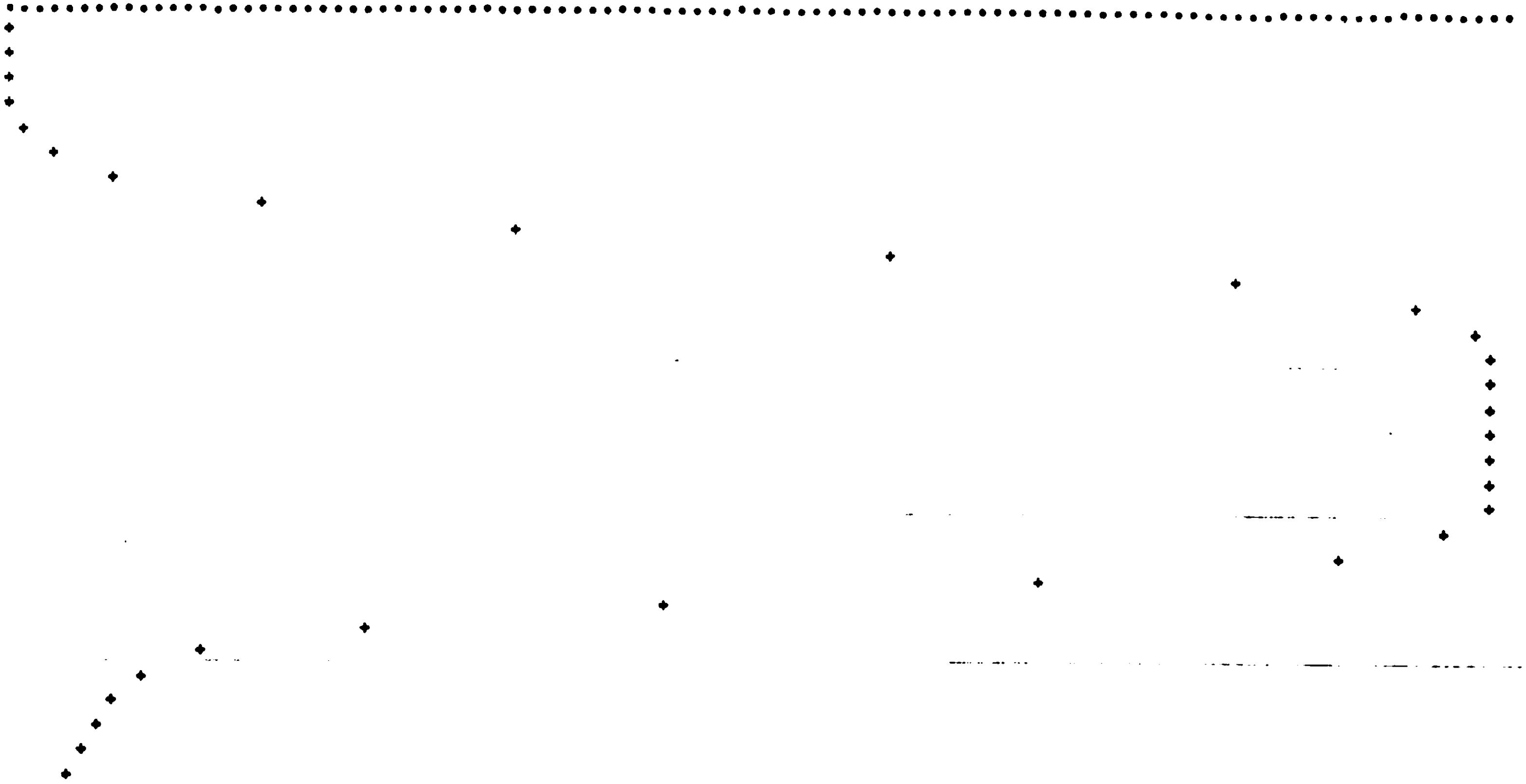
These results agree well with theoretical results for this amplifier.

DATA CARDS: SAMPLE PROBLEM # 6

```
    9   16            1.0        1.0E+09            2.0        1.0E-12         1
    4                            50                        1
E  1       2  1                               0.01
R  2       2  3              1100.0
C  3       3  4             5.0E-06
R  4       4  1              3300.0
R  5       4  1             22000.0
R  6       4  5                50.0
E  7       5  6
R  8       6  7               150.0
R  9       7  8                20.0
R  10      8  1              1000.0
C  11      8  1             5.0E-05
C  12      6  7             1.0E-10
C  13      6  9             1.0E-11
I  14I  7  9  7               100.0
R  15      9  7             30000.0
R  16      9  1              6000.0
    1    1    0
    7    1    0
   14    1    0
   16    1    0
```

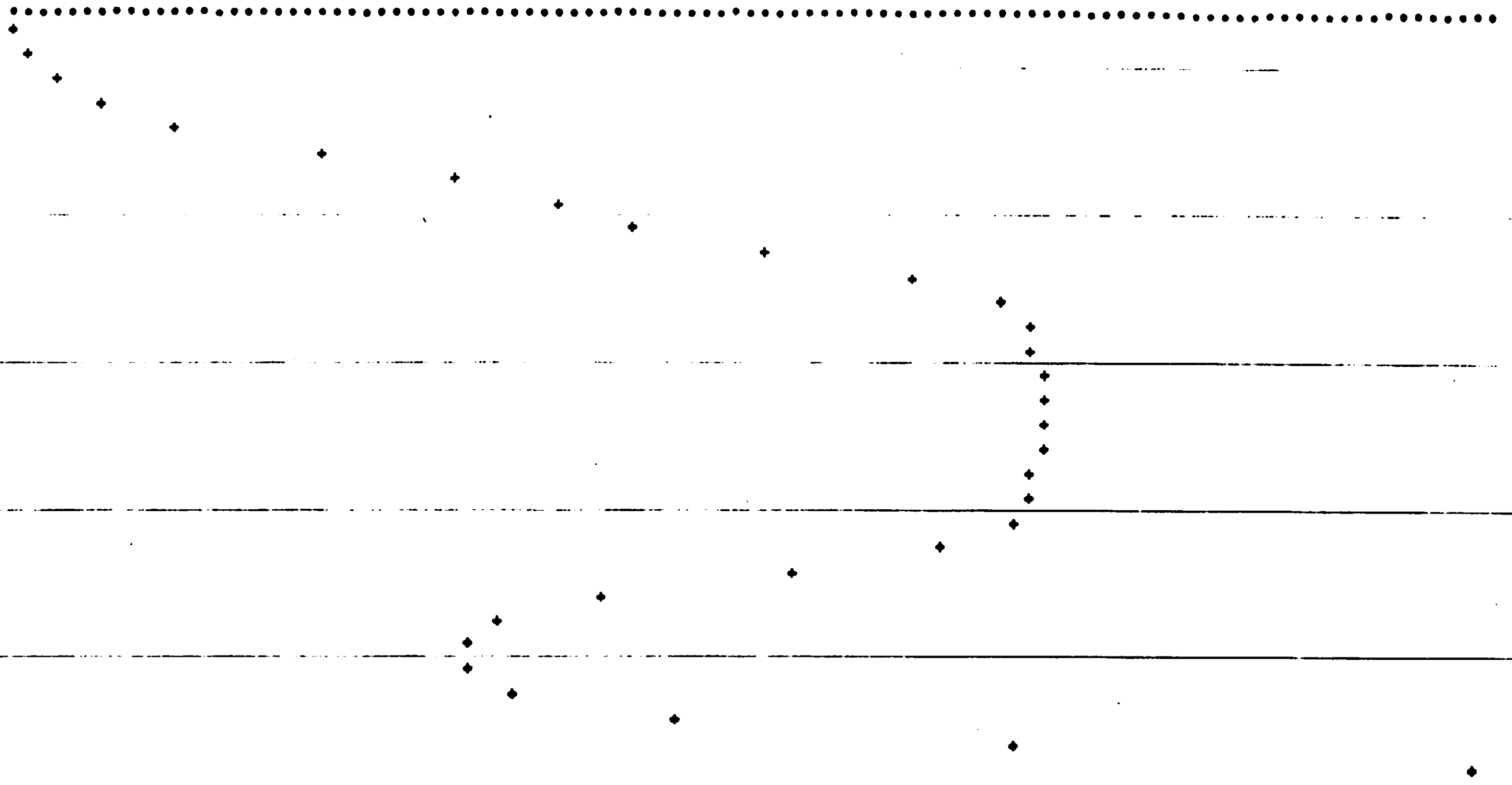
```
                              BRANCH 16     CURRENT      MAGNITUDE PLOT

FREQUENCY       VALUE
 .1000E+01     .1342E-06
 .2000E+01     .2692E-06
 .4000E+01     .5449E-06
 .8000E+01     .1139E-05
 .1600E+02     .2608E-05
 .3200E+02     .6752E-05
 .6400E+02     .1732E-04
 .1280E+03     .3877E-04
 .2560E+03     .7700E-04
 .5120E+03     .1333E-03
 .1024E+04     .1861E-03
 .2048E+04     .2127E-03
 .4096E+04     .2214E-03
 .8192E+04     .2237E-03
 .1638E+05     .2243E-03
 .3277E+05     .2245E-03
 .6554E+05     .2245E-03
 .1311E+06     .2244E-03
 .2621E+06     .2241E-03
 .5243E+06     .2227E-03
 .1049E+07     .2175E-03
 .2097E+07     .1999E-03
 .4194E+07     .1570E-03
 .8389E+07     .9919E-04
 .1678E+08     .5507E-04
 .3355E+08     .3097E-04
 .6711E+08     .2029E-04
 .1342E+09     .1643E-04
 .2684E+09     .1497E-04
 .5369E+09     .1343E-04
 .1074E+10     .1037E-04
```



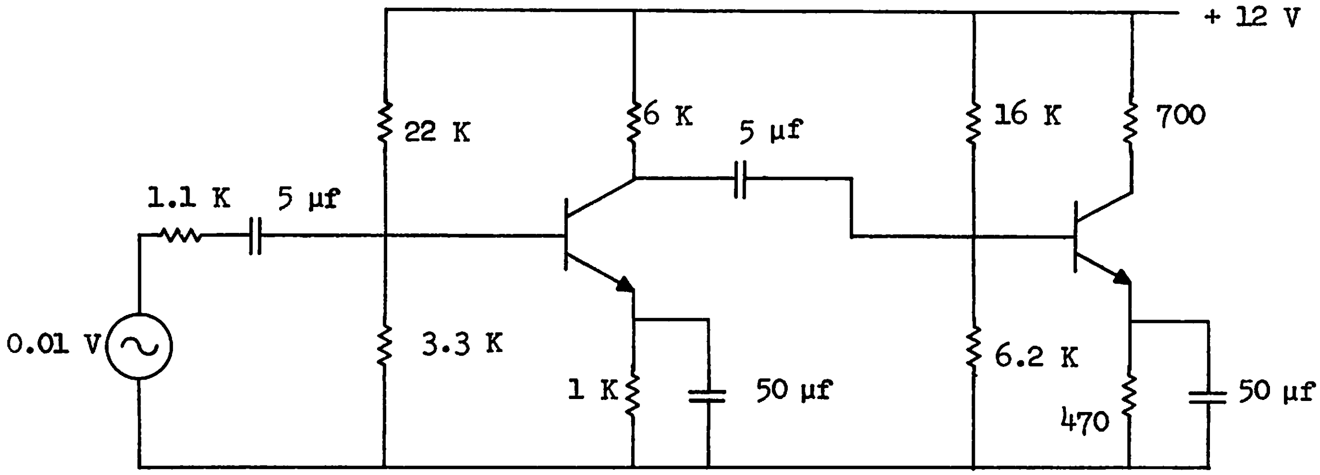
BRANCH 1 CURRENT MAGNITUDE PLOT

```
FREQUENCY     VALUE
 .1000E+01    .4999E-07
 .2000E+01    .9992E-07
 .4000E+01    .1993E-06
 .8000E+01    .3947E-06
 .1600E+02    .7603E-06
 .3200E+02    .1339E-05
 .6400E+02    .1954E-05
 .1280E+03    .2359E-05
 .2560E+03    .2708E-05
 .5120E+03    .3286E-05
 .1024E+04    .3942E-05
 .2048E+04    .4303E-05
 .4096E+04    .4424E-05
 .8192E+04    .4457E-05
 .1638E+05    .4465E-05
 .3277E+05    .4467E-05
 .6554E+05    .4467E-05
 .1311E+06    .4466E-05
 .2621E+06    .4460E-05
 .5243E+06    .4438E-05
 .1049E+07    .4353E-05
 .2097E+07    .4070E-05
 .4194E+07    .3406E-05
 .8389E+07    .2609E-05
 .1678E+08    .2148E-05
 .3355E+08    .2000E-05
 .6711E+08    .2013E-05
 .1342E+09    .2216E-05
 .2684E+09    .2878E-05
 .5369E+09    .4362E-05
 .1074E+10    .6373E-05
```



Sample Problem #7:

Two Stage Audio Amplifier:



We will use the small signal transistor model described in sample problem #6.

Specifications:

- Initial frequency = 1.0 (WSTART = 1.0)
- Final frequency = 10^9 (WEND = 10^9)
- Frequency multiplicative factor = 10^3 (FACTOR = 10^3, FREQADD = 0.0)
- Desire error to fall below 10^{-12} or a maximum of 60 iterative minimization steps at each frequency point. (EPS = 10^{-12}, NITT = 60)
- Do not desire tree and $\hat{F}$ outputed (NCONT = 0)
- Desire an optimal tree to be selected at each frequency point (NTREE = 1)
- Do not desire testing for ideal resonances (IDRES = 1)
- Desire internal, automatic current scaling (SCALE is arbitrary, NSCALE $\neq$ 1)

The central processor time for this problem was 56 seconds.

```
FREQUENCY =       .10000E+01
  BRANCH       1   VOLTAGE       ABS VALUE =        .10000E-01          PHASE =      0.
  BRANCH       1   CURRENT       ABS VALUE =        .49889E-07          PHASE =     -.91107E+02
  BRANCH       7   CURRENT       ABS VALUE =        .16302E-08          PHASE =      .91317E+02
  BRANCH      14   CURRENT       ABS VALUE =        .16302E-06          PHASE =      .91317E+02
  BRANCH      17   CURRENT       ABS VALUE =        .31241E-08          PHASE =      .19030E+00
  BRANCH      20   CURRENT       ABS VALUE =        .10072E-09          PHASE =     -.82742E+02
  BRANCH      22   CURRENT       ABS VALUE =        .42346E-09          PHASE =     -.87670E+02
  BRANCH      25   CURRENT       ABS VALUE =        .10072E-07          PHASE =     -.82794E+02
  BRANCH      30   VOLTAGE       ABS VALUE =        .51371E-05          PHASE =      .10099E+03
  BRANCH      30   CURRENT       ABS VALUE =        .10442E-07          PHASE =      .96154E+02


FREQUENCY =       .10000E+04
  BRANCH       1   VOLTAGE       ABS VALUE =        .10000E-01          PHASE =      0.
  BRANCH       1   CURRENT       ABS VALUE =        .37743E-05          PHASE =     -.16399E+03
  BRANCH       7   CURRENT       ABS VALUE =        .20042E-05          PHASE =      .36456E+02
  BRANCH      14   CURRENT       ABS VALUE =        .20042E-03          PHASE =      .36456E+02
  BRANCH      17   CURRENT       ABS VALUE =        .14709E-03          PHASE =     -.13506E+03
  BRANCH      20   CURRENT       ABS VALUE =        .93781E-04          PHASE =     -.11875E+03
  BRANCH      22   CURRENT       ABS VALUE =        .93780E-04          PHASE =     -.11879E+03
  BRANCH      25   CURRENT       ABS VALUE =        .93781E-02          PHASE =     -.11875E+03
  BRANCH      30   VOLTAGE       ABS VALUE =        .64106E+01          PHASE =      .61285E+02
  BRANCH      30   CURRENT       ABS VALUE =        .91580E-02          PHASE =      .61285E+02
```

```
FREQUENCY =      .10000E+07
  BRANCH          1    VOLTAGE         ABS VALUE =      .10000E-01         PHASE =     0.
  BRANCH          1    CURRENT         ABS VALUE =      .42974E-05         PHASE =    -.17841E+03
  BRANCH          7    CURRENT         ABS VALUE =      .24649E-05         PHASE =     .38127E+01
  BRANCH         14    CURRENT         ABS VALUE =      .24649E-03         PHASE =     .38126E+01
  BRANCH         17    CURRENT         ABS VALUE =      .19051E-03         PHASE =    -.17635E+03
  BRANCH         20    CURRENT         ABS VALUE =      .12821E-03         PHASE =    -.17528E+03
  BRANCH         22    CURRENT         ABS VALUE =      .15619E-03         PHASE =     .14852E+03
  BRANCH         25    CURRENT         ABS VALUE =      .12821E-01         PHASE =    -.17528E+03
  BRANCH         30    VOLTAGE         ABS VALUE =      .87637E+01         PHASE =     .43148E+01
  BRANCH         30    CURRENT         ABS VALUE =      .12520E-01         PHASE =     .43148E+01


FREQUENCY =      .10000E+10
  BRANCH          1    VOLTAGE         ABS VALUE =      .10000E-01         PHASE =     0.
  BRANCH          1    CURRENT         ABS VALUE =      .57985E-05         PHASE =     .13248E+03
  BRANCH          7    CURRENT         ABS VALUE =      .62134E-05         PHASE =    -.71947E+02
  BRANCH         14    CURRENT         ABS VALUE =      .62134E-03         PHASE =    -.71947E+02
  BRANCH         17    CURRENT         ABS VALUE =      .12148E-03         PHASE =    -.68116E+02
  BRANCH         20    CURRENT         ABS VALUE =      .12285E-03         PHASE =    -.76623E+02
  BRANCH         22    CURRENT         ABS VALUE =      .79876E-03         PHASE =     .26153E+02
  BRANCH         25    CURRENT         ABS VALUE =      .12285E-01         PHASE =    -.76623E+02
  BRANCH         30    VOLTAGE         ABS VALUE =      .12951E+01         PHASE =     .22496E+02
  BRANCH         30    CURRENT         ABS VALUE =      .18502E-02         PHASE =     .22495E+02
```

APPENDIX D:

CANDOFD FORTRAN IV LISTING

(CDC 6400)

```
      PROGRAM CANDOFD(INPUT,OUTPUT)
      COMPLEX E,CONST,V,C,GRAD
      COMPLEX CONDTEM
      COMPLEX GRAF
      COMPLEX TEMPAR,DUMMY
      COMPLEX X,Y
      COMPLEX CONT,S
      INTEGER CNTSOR
      INTEGER CONTYPE
      INTEGER CONTTEM
      INTEGER TEMP
      INTEGER TYPE
      COMMON /BLOCK1/ NNP(200),NP(200),IDRES
      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),ITEST(200),NOUT(200),ITTEST(200)
      COMMON /BLOCK6/ TEMP(7),E(200),GRAD(200),CONST(200)
      COMMON /BLOCK7/ ALPHA,FUNCT,NIC,FACTOR,EPS,NN,NB
      COMMON /BLOCK8/ V(200),C(200)
      COMMON /BLOCK9/ CONTYPE(200),KONBRAN(200)
      COMMON /BLOCK10/ AMAG(200),APHASE(200),WSTART,WEND,NRES
      COMMON /BLOCK11/ CONT(200),S(200)
      COMMON /BLOCK12/ CONTTEM(200),CONDTEM(200),KONTEM(200)
      COMMON /BLOCK13/ X(200),Y(200),ITN
      COMMON /BLOCK14/ KP(5),W,FREQADD
      COMMON /BLOCK15/ NIT,JOUT,NGRAPH,NALLOUT
      COMMON /BLOCK16/ GRAF(200,5),JGRAPH(5),IGRAPH(5),NPRINT,SCALE,NITT
      COMMON /BLOCK19/ CNTSOR(200)
      COMMON /BLOCK20/ NCONT,KLOOP,KPP
      COMMON /BLOCK21/ NST(5)
      COMMON /BLOCK23/ TEMPAR(200),DUMMY(200)
      COMMON /BLOCK24/ SCAL,NSCALE,NTREE
      DIMENSION HH(54,54)
      NDATA=0
    1 NIC=0
      NRES=0
      IF (NDATA.EQ.0) GO TO 2
      CALL SECOND (T)
      TO=T-TU
      PRINT 21, TO
    2 CALL SECOND (TU)
      CALL READIN
      NDATA=NDATA+1
      KLOOP=0
      CALL PT
      IF (KLOOP.EQ.1) GO TO 1
      NB2=(NB-NOEL(1)-NOEL(7))*2
      CALL FCSM
      NLS=NN-1
      CALL INITIAL
      NPRINT=0
      IF (FREQADD) 3,4,3
    3 W=WSTART-FREQADD
      GO TO 6
```

```
 4    W=WSTART/FACTOR
 5    W=W*FACTOR
      NTRES=0
      GO TO 7
 6    W=W+FREQADD
      NTRES=0
 7    NIC=NIC+1
      IF (NIC.EQ.1) GO TO 8
      IF (NTREE.NE.1) GO TO 8
      CALL PT
      IF (KPP.EQ.0) GO TO 8
      CALL FCSM
 8    CALL KONST
      NIT=0
      ITN=0
      KRES=0
      IF (NIC.EQ.1) SCAL=1.0
      DO 9 I=NN,NB
      C(I)=C(I)*SCALE/SCAL
 9    CONTINUE
 10   CALL CALCAL
      IF (NITT.EQ.0) GO TO 11
      IF (NIT-NITT) 11,17,17
 11   CALL ERROR
      IF (IDRES.EQ.1) GO TO 13
      IF (KRES.EQ.0) FRES=FUNCT
      IF (KRES.EQ.0) GO TO 12
      A=ABS(FRES-FUNCT)
      B=1.0E-09*FUNCT
      IF (A.LT.B) NRES=1
      FRES=FUNCT
      IF (NTRES.EQ.1) GO TO 13
      IF (NRES.EQ.1) GO TO 19
      GO TO 13
 12   KRES=1
 13   IF (FUNCT-EPS) 17,17,14
 14   CALL FNDGRAD
      ITN=ITN+1
      CALL FLPOW (HH,NB,NB2)
      NIT=NIT+1
      CALL FNDALPH
      CALL CNGVARS
      IF (IDRES.EQ.1) GO TO 10
      DO 15 I=1,NLS
      A=REAL(V(I))
      B=AIMAG(V(I))
      IF (ABS(A).GT.1.0E+30) GO TO 17
      IF (ABS(B).GT.1.0E+30) GO TO 17
 15   CONTINUE
      DO 16 I=NN,NB
      B=AIMAG(C(I))
      A=REAL(C(I))
      IF (ABS(B).GT.1.0E+30) GO TO 17
      IF (ABS(A).GT.1.0E+30) GO TO 17
 16   CONTINUE
```

```
      GO TO 10
   17 NPRINT=NPRINT+1
      IF (NGRAPH.GE.1) CALL GRAPH
      IF (NALLOUT.GE.1) CALL ALLOUT
      IF (JOUT.GE.1) CALL READOUT
      IF (NGRAPH.EQ.201) GO TO 1
      IF (NRES.EQ.1) GO TO 20
      IF (WEND-W) 1,1,18
   18 IF (NIC.EQ.1) CALL INITIAL
      IF (FREQADD) 6,5,6
   19 DW=0.05*W
      W=W+DW
      NIC=NIC-1
      NTRES=1
      GO TO 7
   20 W=W-DW
      NRES=0
      GO TO 18
C
C
   21 FORMAT (1H1,20X,*THE CENTRAL PROCESSOR TIME FOR THIS PROBLEM  =  *
     1,E15.5,*  SECONDS*)
      END
      SUBROUTINE READIN
C     NB IS THE NUMBER OF BRANCHES
C     NN IS THE NUMBER OF NODES
C
      COMPLEX GRAF
      INTEGER CONTYPE
      INTEGER CONTTEM
      INTEGER TYPE
      COMMON /BLOCK1/ NNP(200),NP(200),IDRES
      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),ITEST(200),NOUT(200),ITTEST(200)
      COMMON /BLOCK7/ ALPHA,FUNCT,NIC,FACTOR,EPS,NN,NB
      COMMON /BLOCK9/ CONTYPE(200),KONBRAN(200)
      COMMON /BLOCK10/ AMAG(200),APHASE(200),WSTART,WEND
      COMMON /BLOCK14/ KP(5),W,FREQADD
      COMMON /BLOCK15/ NIT,JOUT,NGRAPH,NALLOUT
      COMMON /BLOCK16/ GRAF(200,5),JGRAPH(5),IGRAPH(5),NPRINT,SCALE,NITT
      COMMON /BLOCK20/ NCONT,KLOOP,KPP
      COMMON /BLOCK24/ SCAL,NSCALE,NTREE
      IJP=0
      READ 12, NN,NB,WSTART,WEND,FACTOR,EPS,NCONT,NTREE
      IF (NN.EQ.0) STOP
      READ 13, NGRAPH,NALLOUT,JOUT,SCALE,NITT,FREQADD,NSCALE,IDRES
      PRINT 14, EPS,WSTART,WEND,FACTOR
      PRINT 15, NN,NB,SCALE,NITT,FREQADD,NSCALE,IDRES
      PRINT 16, NTREE
      DO 1 I=1,NB
      READ 17, TYPE(I),IBRAN(I),CONTYPE(I),KONBRAN(I),LEAV(I),LENT(I),VA
     1LUE(I),AMAG(I),APHASE(I)
 1    CONTINUE
      IF (NGRAPH.EQ.0) GO TO 2
      READ 18, ((IGRAPH(I),JGRAPH(I),ITTEST(I)),I=1,NGRAPH)
 2    IF (NALLOUT.GE.1.OR.NGRAPH.GE.1) GO TO 3
      READ 19, ((IOUT(I),ITEST(I)),I=1,JOUT)
 3    PRINT 20
      PRINT 21
      DO 4 I=1,NB
      PRINT 22, IBRAN(I),TYPE(I),VALUE(I),LEAV(I),LENT(I)
 4    CONTINUE
      PRINT 27
      DO 8 I=1,NB
      IF (TYPE(I).EQ.1HE) GO TO 6
      IF (TYPE(I).EQ.1HJ) GO TO 5
      GO TO 8
 5    A=7HCURRENT
      GO TO 7
 6    A=7HVOLTAGE
 7    PRINT 26, IBRAN(I),A,AMAG(I),APHASE(I)
 8    CONTINUE
      DO 11 I=1,NB
      IF (CONTYPE(I).NE.1H ) GO TO 9
      GO TO 11
 9    IF (IJP.EQ.0) PRINT 23
```

```
      IJP=1
      IF (TYPE(I).EQ.1HI) GO TO 10
      A=7HCURRENT
      IF (CONTYPE(I).EQ.1HV) A=7HVOLTAGE
      PRINT 24, A,IBRAN(I),KONBRAN(I)
      GO TO 11
10    A=7HCURRENT
      IF (CONTYPE(I).EQ.1HV) A=7HVOLTAGE
      PRINT 25, A,IBRAN(I),KONBRAN(I)
11    CONTINUE
      W=WSTART
      RETURN
C
C
12    FORMAT (I5,I5,E15.3,E15.3,2E15.3,2I5)
13    FORMAT (3I5,E15.5,I5,E15.5,I5,I5)
14    FORMAT (1H1,/////,40X,*ERROR CRITERION = *,E15.5,///,40X,* INITIAL
     1 FREQUENCY = *,E15.5, * FINAL FREQUENCY = *, E15.5,///,40X,* FACTO
     2R = *,E15.5)
15    FORMAT (1H0,///,30X,* NUMBER OF NODES = *,I5,*     NUMBER OF BRANC
     1HES = *,I5,///,30X,*   SCALE FACTOR = *,E15.5,///,30X,*DESIRED NUM
     2BER OF ITERATIONS AT EACH FREQUENCY POINT = *,I5,///,30X,*FREQUENC
     3Y ADDITIVE STEP = *,E15.5,///,30X,* NSCALE = *,I5,///,30X,* IDRES
     4 = *,I5)
16    FORMAT (1H0,///,35X,*NTREE = *,I5)
17    FORMAT (A1,I3,1X,A1,I3,1X,I2,1X,I2,5X,3E15.5)
18    FORMAT (3I5)
19    FORMAT (2I5)
20    FORMAT (1H1,55X,*THIS IS THE GIVEN NETWORK*)
21    FORMAT (1H0,///,55X,*UNITS ARE OHMS, FARADS AND HENRYS *)
22    FORMAT (1H0,10X,*BRANCH NUMBER *,5X,I3,4X,* IS A      *,A1,*     OF
     1 VALUE  *,E12.5,*  LEAVING NODE   *,I3,*  AND ENTERING NODE  *,I3)
23    FORMAT (1H0,///,30X,*CONTROLLED SOURCES *,///)
24    FORMAT (1H0,10X,A7,* CONTROLLED VOLTAGE SOURCE*,I3,* IS CONTROLLED
     1 BY BRANCH*,I5)
25    FORMAT (1H0,10X,A7,* CONTROLLED CURRENT SOURCE*,I3,* IS CONTROLLED
     1 BY BRANCH*,I5)
26    FORMAT (1H0,10X,*BRANCH*,I4,3X,* IS AN INDEPENDENT *,A7,* SOURCE*,
     110X,*MAGNITUDE = *,E15.5,10X,*PHASE = *,E15.5)
27    FORMAT (1H1,30X,*INDEPENDENT SOURCES*,///)
      END
      SUBROUTINE PT
      COMPLEX V,C
      COMPLEX TEMP11,TEMP12
      COMPLEX GRAF
      INTEGER CNTSOR
      INTEGER CONTTEM
      INTEGER TYTEMP
      INTEGER TEMP01,TEMP02,TEMP03,TEMP04,TEMP06,TEMP09
      INTEGER TEMP
      INTEGER TYPE
      INTEGER CONTYPE
      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),ITEST(200),NOUT(200),ITTEST(200)
      COMMON /BLOCK6/ TEMP(7),ITBB(200),TYTEMP(200),VALTEM(200)
      COMMON /BLOCK7/ ALPHA,FUNCT,NIC,FACTOR,EPS,NN,NB
      COMMON /BLOCK8/ V(200),C(200)
      COMMON /BLOCK9/ CONTYPE(200),KONBRAN(200)
      COMMON /BLOCK10/ AMAG(200),APHASE(200),WSTART,WEND
      COMMON /BLOCK12/ CONTTEM(200),AMTEM(200),APH(200),KONTEM(200)
      COMMON /BLOCK14/ KP(5),W,FREQADD
      COMMON /BLOCK15/ NIT,JOUT,NGRAPH,NALLOUT
      COMMON /BLOCK16/ GRAF(200,5),JGRAPH(5),IGRAPH(5),NPRINT,SCALE,NITT
      COMMON /BLOCK19/ CNTSOR(200)
      COMMON /BLOCK20/ NCONT,KLOOP,KPP
      COMMON /BLOCK21/ NST(5)
      COMMON /BLOCK23/ CONST(200)
      COMMON /BLOCK24/ SCAL,NSCALE,NTREE
      KK=NN-1
      KKTT=0
      IF (NIC.EQ.0) GO TO 1
      GO TO 6
1     KM=0
      TEMP(1)=1HE
      TEMP(2)=1HV
      TEMP(3)=1HC
      TEMP(4)=1HR
      TEMP(5)=1HL
      TEMP(6)=1HI
      TEMP(7)=1HJ
      DO 4 K=1,7
      KT=0
      DO 3 I=1,NB
      IF (TYPE(I).EQ.TEMP(K)) GO TO 2
      GO TO 3
2     KM=KM+1
      ITBRAN(KM)=I
      TYTEMP(KM)=TEMP(K)
      VALTEM(KM)=VALUE(I)
      KT=KT+1
      CONTTEM(KM)=CONTYPE(I)
      AMTEM(KM)=AMAG(I)
      APH(KM)=APHASE(I)
      KONTEM(KM)=KONBRAN(I)
    3 CONTINUE
      NOEL(K)=KT
    4 CONTINUE
      DO 5 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)
      AMAG(I)=AMTEM(I)
      APHASE(I)=APH(I)
      KONBRAN(I)=KONTEM(I)
    5 CONTINUE
      MM=NB
      N=1
      GO TO 7
    6 MM=NB-NOEL(7)
      N=NOEL(1)+1
    7 M=MM-1
      DO 8 I=N,MM
      IF (TYPE(I).EQ.1HE) CONST(I)=-1.0
      IF (TYPE(I).EQ.1HV) CONST(I)=1.0E-50
      IF (TYPE(I).EQ.1HC) GO TO 9
      IF (TYPE(I).EQ.1HR) CONST(I)=VALUE(I)
      IF (TYPE(I).EQ.1HL) CONST(I)=VALUE(I)*W
      IF (TYPE(I).EQ.1HI) CONST(I)=1.0E+50
      IF (TYPE(I).EQ.1HJ) CONST(I)=1.0E+52
    8 CONTINUE
      GO TO 10
    9 A=VALUE(I)*W
      IF (A.EQ.0.0) CONST(I)=1.0E+51
      IF (A.NE.0.0) CONST(I)=1.0/A
      GO TO 8
   10 KPP=0
      IF (N.GT.M) GO TO 16
      DO 15 I=N,M
      KP=0
      AMIN=CONST(I)
      K=I+1
      DO 12 J=K,MM
      IF (CONST(J).LT.AMIN) GO TO 11
      GO TO 12
   11 AMIN=CONST(J)
      NP(I)=J
      KP=1
      KPP=1
   12 CONTINUE
      IF (KP.EQ.0) GO TO 15
      J=NP(I)
      TEMPO1=ITBRAN(I)
      TEMPO2=LEAVT(I)
      TEMPO3=LENTT(I)
      TEMPO4=TYPE(I)
      TEMPO5=VALUE(I)
      TEMPO6=CONTYPE(I)
      TEMP20=AMAG(I)
      TEMP21=APHASE(I)
      TEMPO9=KONBRAN(I)
      TEMP10=CONST(I)
      IF (NIC.EQ.0) GO TO 13
      TEMP11=V(I)
      TEMP12=C(I)
 13   ITBRAN(I)=ITBRAN(J)
      LEAVT(I)=LEAVT(J)
      LENTT(I)=LENTT(J)
      TYPE(I)=TYPE(J)
      VALUE(I)=VALUE(J)
      CONTYPE(I)=CONTYPE(J)
      AMAG(I)=AMAG(J)
      APHASE(I)=APHASE(J)
      KONBRAN(I)=KONBRAN(J)
      CONST(I)=CONST(J)
      IF (NIC.EQ.0) GO TO 14
      V(I)=V(J)
      C(I)=C(J)
 14   ITBRAN(J)=TEMPO1
      LEAVT(J)=TEMPO2
      LENTT(J)=TEMPO3
      TYPE(J)=TEMPO4
      VALUE(J)=TEMPO5
      CONTYPE(J)=TEMPO6
      AMAG(J)=TEMP20
      APHASE(J)=TEMP21
      KONBRAN(J)=TEMPO9
      CONST(J)=TEMP10
      IF (NIC.EQ.0) GO TO 15
      V(J)=TEMP11
      C(J)=TEMP12
 15   CONTINUE
 16   IF (2.GT.KK) GO TO 32
      DO 26 I=2,KK
 17   NE=LENTT(I)
      NL=LEAVT(I)
      NP(1)=I
      NNP(1)=1
      M=I-1
      MT=0
      KT=1
 18   DO 20 JJJ=1,M
      J=M+1-JJJ
      IF (MT.EQ.J) GO TO 19
      IF (LEAVT(J).EQ.NE) GO TO 24
      IF (LENTT(J).EQ.NE) GO TO 25
 19   CONTINUE
 20   CONTINUE
 21   IF (KT.EQ.1) GO TO 26
```

```
      KA=NNP(KT)
      KB=NP(KT)
      M=KB-1
      IF (M.EQ.0) GO TO 22
      MT=NP(KT-1)
      KT=KT-1
      IF (KA.EQ.-1) GO TO 23
      NE=LEAVT(KB)
      GO TO 18
   22 KT=KT-1
      GO TO 21
   23 NE=LENTT(KB)
      GO TO 18
   24 IF (LENTT(J).EQ.NL) GO TO 27
      KT=KT+1
      NE=LENTT(J)
      NP(KT)=J
      NNP(KT)=1
      MT=J
      M=I-1
      GO TO 18
   25 IF (LEAVT(J).EQ.NL) GO TO 27
      NE=LEAVT(J)
      KT=KT+1
      NP(KT)=J
      NNP(KT)=-1
      MT=J
      M=I-1
      GO TO 18
   26 CONTINUE
      GO TO 32
   27 LAF=NOEL(1)
      IF (I.LE.LAF) GO TO 52
   28 TEMPO1=ITBRAN(I)
      TEMPO2=LEAVT(I)
      TEMPO3=LENTT(I)
      TEMPO4=TYPE(I)
      TEMPO5=VALUE(I)
      TEMPO6=CONTYPE(I)
      TEMP20=AMAG(I)
      TEMP21=APHASE(I)
      TEMPO9=KONBRAN(I)
      IF (NIC.EQ.0) GO TO 29
      TEMP11=V(I)
      TEMP12=C(I)
   29 NLB=NB-1
      IF (NIC.NE.0) NLB=NB-NOEL(7)-1
      DO 30 JN=I,NLB
      JE=JN+1
      ITBRAN(JN)=ITBRAN(JE)
      LEAVT(JN)=LEAVT(JE)
      LENTT(JN)=LENTT(JE)
      TYPE(JN)=TYPE(JE)
      VALUE(JN)=VALUE(JE)
      CONTYPE(JN)=CONTYPE(JE)
      AMAG(JN)=AMAG(JE)
      APHASE(JN)=APHASE(JE)
      KONBRAN(JN)=KONBRAN(JE)
      IF (NIC.EQ.0) GO TO 30
      V(JN)=V(JE)
      C(JN)=C(JE)
   30 CONTINUE
      NZ=NB
      IF (NIC.NE.0) NZ=NB-NOEL(7)
      ITBRAN(NZ)=TEMP01
      LEAVT(NZ)=TEMP02
      LENTT(NZ)=TEMP03
      TYPE(NZ)=TEMP04
      VALUE(NZ)=TEMP05
      CONTYPE(NZ)=TEMP06
      AMAG(NZ)=TEMP20
      APHASE(NZ)=TEMP21
      KONBRAN(NZ)=TEMP09
      IF (NIC.EQ.0) GO TO 31
      V(NZ)=TEMP11
      C(NZ)=TEMP12
   31 IF (KKTT.EQ.1) GO TO 62
      GO TO 17
   32 CONTINUE
      IF (NIC.NE.0) GO TO 36
      DO 33 L=1,KK
      IF (TYPE(L).EQ.TEMP(7)) GO TO 53
   33 CONTINUE
      I=NN-1
      DO 35 J=NN,NB
      I=I+1
      IF (TYPE(I).EQ.TEMP(7)) GO TO 34
      GO TO 35
   34 KKTT=1
      GO TO 28
   62 I=I-1
   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 56
      PRINT 57, ((I,NP(I)),I=1,KK)
      PRINT 58, ((I,NP(I)),I=NN,NB)
      PRINT 59
      DO 51 I=1,NB
      PRINT 55, NP(I),TYPE(I),VALUE(I),LEAVT(I),LENTT(I)
 51   CONTINUE
      RETURN
 52   PRINT 60
      KLOOP=1
      RETURN
 53   PRINT 61
      KLOOP=1
 54   RETURN
C
C
 55   FORMAT (1H0,10X,*BRANCH NUMBER *,5X,I3,4X,* IS A     *,A1,*     OF
     1 VALUE  *,E12.5,*  LEAVING NODE   *,I3,*  AND ENTERING NODE  *,I3)
 56   FORMAT (1H1,20X,*CORRESPONDENCE BETWEEN ORIGINAL TOPOLOGY AND NEW
     1TOPOLOGY*,///)
 57   FORMAT (1H0,30X,*TREE BRANCH*,I4,5X,*CORRESPONDS TO BRANCH *,I4)
 58   FORMAT (1H0,33X,* LINK *,2X,I4,5X,*CORRESPONDS TO BRANCH *,I4)
 59   FORMAT (1H1)
     12X,* COND = *,E10.3)
 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 TYPE
      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 /BLOCK7/ ALPHA,FUNCT,NIC,FACTOR,EPS,NN,NB
      COMMON /BLOCK20/ NCONT,KLOOP,KPP
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 20, NPAGE
      PRINT 19
      PRINT 18, (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 17, 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 17, I,(F(I,J),J=NS3,NS4)
 15   CONTINUE
 16   RETURN
C
C
 17   FORMAT (1H0,I3,3X,25(2X,F3.0))
 18   FORMAT (1H0,7X,25(2X,I2,1X))
 19   FORMAT (///)
 20   FORMAT (1H1,30X,*  F PORTION OF FUNDAMENTAL CUTSET MATRIX*,5X,*PAG
     1E*,I5,///)
      END
      SUBROUTINE VOLCUR (I,D)
      COMPLEX D
      COMMON /BLOCK10/ AMAG(200),APHASE(200),WSTART,WEND
      PI=180.0/3.14159265
      IF (APHASE(I).LT.-180.) APHASE(I)=360.0+APHASE(I)
      IF (APHASE(I).GT.180.) APHASE(I)=-360.+APHASE(I)
      IF (APHASE(I)) 1,5,6
    1 IF (APHASE(I)+90.0) 2,3,4
    2 TH=(180.0+APHASE(I))/PI
      A=-COS(TH)*AMAG(I)
      B=-SIN(TH)*AMAG(I)
      D=CMPLX(A,B)
      RETURN
    3 B=-AMAG(I)
      D=CMPLX(0.0,B)
      RETURN
    4 TH=-APHASE(I)/PI
      A=COS(TH)*AMAG(I)
      B=-SIN(TH)*AMAG(I)
      D=CMPLX(A,B)
      RETURN
    5 A=AMAG(I)
      D=CMPLX(A,0.0)
      RETURN
    6 IF (APHASE(I)-90.0) 7,8,9
    7 TH=APHASE(I)/PI
      A=COS(TH)*AMAG(I)
      B=SIN(TH)*AMAG(I)
      D=CMPLX(A,B)
      RETURN
    8 B=AMAG(I)
      D=CMPLX(0.0,B)
      RETURN
    9 TH=(180.0-APHASE(I))/PI
      A=-AMAG(I)*COS(TH)
      B=AMAG(I)*SIN(TH)
      D=CMPLX(A,B)
      RETURN
      END
      SUBROUTINE INITIAL
      COMPLEX V,C
      COMPLEX GRAF
      INTEGER TYPE
      COMMON /BLOCK3/ TYPE(200),VALUE(200),F(49,151),NOEL(7)
      COMMON /BLOCK7/ ALPHA,FUNCT,NIC,FACTOR,EPS,NN,NB
      COMMON /BLOCK8/ V(200),C(200)
      COMMON /BLOCK10/ AMAG(200),APHASE(200),WSTART,WEND
      COMMON /BLOCK16/ GRAF(200,5),JGRAPH(5),IGRAPH(5),NPRINT,SCALE,NITT
      N=NN-1
      IF (NIC.EQ.0) GO TO 4
      IF (WSTART.NE.0.0) RETURN
    1 DO 2 I=1,N
      V(I)=CMPLX(0.0,0.0)
      IF (TYPE(I).EQ.1HE) CALL VOLCUR (I,V(I))
    2 CONTINUE
      DO 3 I=NN,NB
      C(I)=CMPLX(0.0,0.0)
      IF (TYPE(I).EQ.1HJ) CALL VOLCUR (I,C(I))
      IF (NIC.GT.0) C(I)=C(I)*SCALE
    3 CONTINUE
      RETURN
    4 IF (WSTART.NE.0.0) GO TO 1
      DO 5 I=1,N
      V(I)=CMPLX(0.0,0.0)
      IF (TYPE(I).EQ.1HE) V(I)=CMPLX(AMAG(I),0.0)
    5 CONTINUE
      DO 6 I=NN,NB
      C(I)=CMPLX(0.0,0.0)
      IF (TYPE(I).EQ.1HJ) C(I)=CMPLX(AMAG(I),0.0)
    6 CONTINUE
      RETURN
      END
      SUBROUTINE CALCAL
      COMPLEX E,CONST,V,C,GRAD
      INTEGER TYPE
      COMMON /BLOCK3/ TYPE(200),VALUE(200),F(49,151),NOEL(7)
      COMMON /BLOCK7/ ALPHA,FUNCT,NIC,FACTOR,EPS,NN,NB
      COMMON /BLOCK8/ V(200),C(200)
      N=NN-1
      DO 1 I=1,N
      C(I)=CMPLX(0.0,0.0)
      DO 1 J=NN,NB
      C(I)=C(I)-F(I,J-N)*C(J)
1     CONTINUE
      DO 2 I=NN,NB
      V(I)=CMPLX(0.0,0.0)
      DO 2 J=1,N
      V(I)=V(I)+F(J,I-N)*V(J)
2     CONTINUE
      RETURN
      END
      SUBROUTINE KONST
C     CALCULATION OF CONSTANTS RELATED TO ELEMENT VALUES FOR USE IN BRAN
C     RELATIONS AND GRADIENT
      COMPLEX E,CONST,V,C,GRAD
      COMPLEX GRAF
      INTEGER CONTYPE
      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 /BLOCK7/ ALPHA,FUNCT,NIC,FACTOR,EPS,NN,NB
      COMMON /BLOCK8/ V(200),C(200)
      COMMON /BLOCK9/ CONTYPE(200),KONBRAN(200)
      COMMON /BLOCK14/ KP(5),W,FREQADD
      COMMON /BLOCK16/ GRAF(200,5),JGRAPH(5),IGRAPH(5),NPRINT,SCALE,NITT
      COMMON /BLOCK24/ SCAL,NSCALE,NTREE
      SCAL=SCALE
      IF (NSCALE.EQ.1) GO TO 3
      KW=0
      DO 2 I=NN,NB
      IF (TYPE(I).EQ.1HJ) GO TO 2
      IF (TYPE(I).EQ.1HV) GO TO 2
      IF (TYPE(I).EQ.1HI) GO TO 2
      IF (TYPE(I).EQ.1HC) A=W*VALUE(I)
      IF (TYPE(I).EQ.1HR) A=1.0/VALUE(I)
      IF (W.EQ.0.0) GO TO 1
      IF (TYPE(I).EQ.1HL) A=1.0/(W*VALUE(I))
 1    IF (KW.EQ.0) SCALE=A
      KW=1
      IF (A.GT.SCALE) SCALE=A
 2    CONTINUE
      IF (KW.EQ.0) SCALE=1.0
      IF (SCALE.EQ.0.0) SCALE=1.0
      SCALE=1.0/SCALE
 3    N=NN-1
      DO 9 I=1,N
      IF (TYPE(I).EQ.1HV) GO TO 4
      IF (TYPE(I).EQ.1HC) GO TO 5
      IF (TYPE(I).EQ.1HR) GO TO 6
      IF (TYPE(I).EQ.1HL) GO TO 7
      IF (TYPE(I).EQ.1HI) GO TO 8
      GO TO 9
 4    CONST(I)=CMPLX(VALUE(I),0.0)
      IF (CONTYPE(I).EQ.1HI) CONST(I)=CONST(I)/SCALE
      GO TO 9
 5    IF (W.EQ.0.0) GO TO 9
      A=-1.0/(VALUE(I)*SCALE*W)
      CONST(I)=CMPLX(0.0,A)
      GO TO 9
 6    A=VALUE(I)/SCALE
      CONST(I)=CMPLX(A,0.0)
      GO TO 9
 7    A=VALUE(I)*W/SCALE
      CONST(I)=CMPLX(0.0,A)
      GO TO 9
```

```
  8   CONST(I)=CMPLX(VALUE(I),0.0)
      IF (CONTYPE(I).EQ.1HV) CONST(I)=CONST(I)*SCALE
  9   CONTINUE
      DO 15 I=NN,NB
      IF (TYPE(I).EQ.1HV) GO TO 10
      IF (TYPE(I).EQ.1HC) GO TO 11
      IF (TYPE(I).EQ.1HR) GO TO 12
      IF (TYPE(I).EQ.1HL) GO TO 13
      IF (TYPE(I).EQ.1HI) GO TO 14
      GO TO 15
 10   CONST(I)=CMPLX(VALUE(I),0.0)
      IF (CONTYPE(I).EQ.1HI) CONST(I)=CONST(I)/SCALE
      GO TO 15
 11   A=VALUE(I)*SCALE*W
      CONST(I)=CMPLX(0.0,A)
      GO TO 15
 12   A=SCALE/VALUE(I)
      CONST(I)=CMPLX(A,0.0)
      GO TO 15
 13   IF (W.EQ.0.0) GO TO 15
      A=-SCALE/(VALUE(I)*W)
      CONST(I)=CMPLX(0.0,A)
      GO TO 15
 14   CONST(I)=CMPLX(VALUE(I),0.0)
      IF (CONTYPE(I).EQ.1HV) CONST(I)=CONST(I)*SCALE
 15   CONTINUE
      RETURN
      END
      SUBROUTINE ERROR
      INTEGER CNTSOR
      COMPLEX E,CONST,V,C,GRAD
      INTEGER TEMP
      INTEGER CONTYPE
      INTEGER TYPE
      COMMON /BLOCK3/ TYPE(200),VALUE(200),F(49,151),NOEL(7)
      COMMON /BLOCK6/ TEMP(7),E(200),GRAD(200),CONST(200)
      COMMON /BLOCK7/ ALPHA,FUNCT,NIC,FACTOR,EPS,NN,NB
      COMMON /BLOCK8/ V(200),C(200)
      COMMON /BLOCK9/ CONTYPE(200),KONBRAN(200)
      COMMON /BLOCK14/ KP(5),W,FREQADD
      COMMON /BLOCK19/ CNTSOR(200)
      M=NOEL(1)+1
      N=NB-NOEL(7)
      DO 10 I=M,N
      IF (TYPE(I).EQ.1HV) GO TO 8
      IF (TYPE(I).EQ.1HI) GO TO 6
      IF (I.GE.NN) GO TO 3
      IF (W.EQ.0.0) GO TO 2
 1    E(I)=-V(I)+C(I)*CONST(I)
      GO TO 10
 2    IF (TYPE(I).NE.1HC) GO TO 1
      E(I)=CMPLX(0.0,0.0)
      V(I)=CMPLX(0.0,0.0)
      GO TO 10
 3    IF (W.EQ.0.0) GO TO 5
 4    E(I)=-C(I)+V(I)*CONST(I)
      GO TO 10
 5    IF (TYPE(I).NE.1HL) GO TO 4
      E(I)=CMPLX(0.0,0.0)
      C(I)=CMPLX(0.0,0.0)
      GO TO 10
 6    K=CNTSOR(I)
      IF (CONTYPE(I).NE.1HV) GO TO 7
      E(I)=-C(I)+V(K)*CONST(I)
      GO TO 10
 7    E(I)=-C(I)+C(K)*CONST(I)
      GO TO 10
 8    K=CNTSOR(I)
      IF (CONTYPE(I).NE.1HV) GO TO 9
      E(I)=-V(I)+V(K)*CONST(I)
      GO TO 10
 9    E(I)=-V(I)+C(K)*CONST(I)
 10   CONTINUE
      FUNCT=0.0
      DO 11 I=M,N
      FUNCT=FUNCT+E(I)*CONJG(E(I))
 11   CONTINUE
      FUNCT=FUNCT/2
      RETURN
      END
      SUBROUTINE CNGVARS
C     THIS SUBROUTINE CHANGES THE CURRENTS ALONG THE GRADIENT BY THE
C     CONSTANT ALPHA
```

```
      COMPLEX V,C
      COMPLEX CONT,S
      INTEGER TYPE
      COMMON /BLOCK3/ TYPE(200),VALUE(200),F(49,151),NOEL(7)
      COMMON /BLOCK7/ ALPHA,FUNCT,NIC,FACTOR,EPS,NN,NB
      COMMON /BLOCK8/ V(200),C(200)
      COMMON /BLOCK11/ CONT(200),S(200)
      M=NOEL(1)+1
      N=NB-NOEL(7)
      IEND=NN-1
      IF (M.GT.IEND) GO TO 2
      DO 1 I=M,IEND
      V(I)=V(I)-ALPHA*S(I)
 1    CONTINUE
 2    IF (NN.GT.N) RETURN
      DO 3 I=NN,N
      C(I)=C(I)-ALPHA*S(I)
 3    CONTINUE
      RETURN
      END
      SUBROUTINE FNDGRAD
C     .      THIS SUBROUTINE CALCULATES THE GRADIENT VECTOR
      INTEGER CNTSOR
      COMPLEX TEMPAR,DUMMY
      COMPLEX E,CONST,V,C,GRAD
      INTEGER CONTYPE
      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 /BLOCK7/ ALPHA,FUNCT,NIC,FACTOR,EPS,NN,NB
      COMMON /BLOCK8/ V(200),C(200)
      COMMON /BLOCK9/ CONTYPE(200),KONBRAN(200)
      COMMON /BLOCK14/ KP(5),W,FREQADD
      COMMON /BLOCK19/ CNTSOR(200)
      COMMON /BLOCK23/ TEMPAR(200),DUMMY(200)
      M=NOEL(1)+1
      MM=NB-NOEL(7)
      N=NN-1
      DO 7 I=1,N
      IF (TYPE(I).EQ.1HE) GO TO 3
      IF (TYPE(I).EQ.1HI) GO TO 5
      IF (TYPE(I).EQ.1HV) GO TO 6
      IF (W.EQ.0.0) GO TO 2
    1 TEMPAR(I)=-E(I)
      A=REAL(E(I))*REAL(CONST(I))+AIMAG(E(I))*AIMAG(CONST(I))
      B=-REAL(E(I))*AIMAG(CONST(I))+AIMAG(E(I))*REAL(CONST(I))
      DUMMY(I)=CMPLX(A,B)
      GO TO 7
    2 IF (TYPE(I).NE.1HC) GO TO 1
      TEMPAR(I)=CMPLX(0.0,0.0)
      DUMMY(I)=CMPLX(0.0,0.0)
      GO TO 7
    3 K=CNTSOR(I)
      IF (K.EQ.0) GO TO 4
      TEMPAR(I)=CMPLX(0.0,0.0)
      DUMMY(I)=CONST(K)*E(K)
      GO TO 7
    4 TEMPAR(I)=CMPLX(0.0,0.0)
      DUMMY(I)=CMPLX(0.0,0.0)
      GO TO 7
    5 TEMPAR(I)=CMPLX(0.0,0.0)
      DUMMY(I)=-E(I)
      GO TO 7
    6 TEMPAR(I)=-E(I)
      DUMMY(I)=CMPLX(0.0,0.0)
    7 CONTINUE
      DO 14 I=NN,NB
      IF (TYPE(I).EQ.1HJ) GO TO 10
      IF (TYPE(I).EQ.1HV) GO TO 12
      IF (TYPE(I).EQ.1HI) GO TO 13
      IF (W.EQ.0.0) GO TO 9
    8 DUMMY(I)=-E(I)
      A=REAL(E(I))*REAL(CONST(I))+AIMAG(E(I))*AIMAG(CONST(I))
      B=-REAL(E(I))*AIMAG(CONST(I))+AIMAG(E(I))*REAL(CONST(I))
      TEMPAR(I)=CMPLX(A,B)
      GO TO 14
    9 IF (TYPE(I).NE.1HL) GO TO 8
      TEMPAR(I)=CMPLX(0.0,0.0)
      DUMMY(I)=CMPLX(0.0,0.0)
      GO TO 14
   10 K=CNTSOR(I)
      IF (K.EQ.0) GO TO 11
      TEMPAR(I)=CONST(K)*E(K)
      DUMMY(I)=CMPLX(0.0,0.0)
      GO TO 14
   11 TEMPAR(I)=CMPLX(0.0,0.0)
      DUMMY(I)=CMPLX(0.0,0.0)
      GO TO 14
   12 TEMPAR(I)=-E(I)
      DUMMY(I)=CMPLX(0.0,0.0)
      GO TO 14
   13 TEMPAR(I)=CMPLX(0.0,0.0)
      DUMMY(I)=-E(I)
   14 CONTINUE
      IF (M.GT.N) GO TO 16
      DO 15 I=M,N
      GRAD(I)=TEMPAR(I)
      DO 15 J=NN,NB
   15 GRAD(I)=GRAD(I)+F(I,J-N)*TEMPAR(J)
   16 IF (NN.GT.MM) GO TO 18
      DO 17 I=NN,MM
      GRAD(I)=DUMMY(I)
      DO 17 J=1,N
   17 GRAD(I)=GRAD(I)-F(J,I-N)*DUMMY(J)
   18 IF (W.EQ.0.0) GO TO 19
      RETURN
   19 IF (M.GT.N) GO TO 21
      DO 20 I=M,N
      IF (TYPE(I).EQ.1HC) GRAD(I)=CMPLX(0.0,0.0)
   20 CONTINUE
   21 IF (NN.GT.MM) RETURN
      DO 22 I=NN,MM
      IF (TYPE(I).EQ.1HL) GRAD(I)=CMPLX(0.0,0.0)
   22 CONTINUE
      RETURN
      END
      SUBROUTINE FNDALPH
C     THIS SUBROUTINE FINDS THE DISTANCE WE SHOULD GO ALOND THE GRADIENT
      COMPLEX E,CONST,V,C,GRAD
      COMPLEX CONT,S
      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/ ALPHA,FUNCT,NIC,FACTOR,EPS,NN,NB
      COMMON /BLOCK8/ V(200),C(200)
      COMMON /BLOCK11/ CONT(200),S(200)
      FZERO=FUNCT
      M=NOEL(1)+1
      N=NB-NOEL(7)
      MM=NN-1
      PROD=0.0
      IF (M.GT.MM) GO TO 2
      DO 1 I=M,MM
      PROD=PROD+REAL(GRAD(I))*REAL(S(I))+AIMAG(GRAD(I))*AIMAG(S(I))
      CONT(I)=V(I)
      V(I)=V(I)-S(I)
  1   CONTINUE
  2   IF (NN.GT.N) GO TO 4
      DO 3 I=NN,N
      PROD=PROD+REAL(GRAD(I))*REAL(S(I))+AIMAG(GRAD(I))*AIMAG(S(I))
      CONT(I)=C(I)
      C(I)=C(I)-S(I)
  3   CONTINUE
  4   CALL CALCAL
      CALL ERROR
      FGRAD=FUNCT
      IF (M.GT.MM) GO TO 6
      DO 5 I=M,MM
      V(I)=CONT(I)
  5   CONTINUE
  6   IF (NN.GT.N) GO TO 8
      DO 7 I=NN,N
      C(I)=CONT(I)
  7   CONTINUE
  8   A=PROD+FGRAD-FZERO
      IF (A.EQ.0.0) A=1.0E-10
      ALPHA=PROD/A
      ALPHA=ALPHA/2.0
      RETURN
      END
      SUBROUTINE FLPOW (HH,NB,NB2)
      COMPLEX CONT,S
      COMPLEX E,GRAD,CONST
      COMPLEX GRADB
      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 /BLOCK7/ ALPHA
      COMMON /BLOCK11/ CONT(200),S(200)
      COMMON /BLOCK13/ GRADB(200),DUMMY(400),ITN
      COMMON /BLOCK23/ SS(400),GB(400)
      DIMENSION HH(NB2,NB2)
      IF (ITN.GT.1) GO TO 3
      M=NOEL(1)
      NB1=NB2/2
      DO 2 I=1,NB1
      II=I+NB1
      GRADB(I+M)=GRAD(I+M)
      DO 1 J=1,NB2
      HH(I,J)=0.0
    1 HH(II,J)=0.0
      HH(II,II)=1.0
      HH(I,I)=1.0
    2 S(I+M)=GRAD(I+M)
      RETURN
    3 GHG=0.0
      SG=0.0
      DO 4 I=1,NB1
      II=I+NB1
      L=I+M
      SS(I)=REAL(S(L))*ALPHA
      SS(II)=AIMAG(S(L))*ALPHA
      GB(I)=REAL(GRAD(L))-REAL(GRADB(L))
      GB(II)=AIMAG(GRAD(L))-AIMAG(GRADB(L))
    4 SG=SG+SS(I)*GB(I)+SS(II)*GB(II)
      DO 5 I=1,NB2
      DUMMY(I)=0.0
      DO 5 J=1,NB2
      GHG=GHG+GB(I)*GB(J)*HH(I,J)
    5 DUMMY(I)=DUMMY(I)+HH(I,J)*GB(J)
      DO 6 I=1,NB2
      DO 6 J=I,NB2
      HH(I,J)=HH(I,J)-SS(I)*SS(J)/SG-DUMMY(I)*DUMMY(J)/GHG
    6 HH(J,I)=HH(I,J)
      DO 7 I=1,NB1
      II=I+NB1
      L=I+M
      GB(I)=REAL(GRAD(L))
    7 GB(II)=AIMAG(GRAD(L))
      DO 8 I=1,NB2
      SS(I)=0.0
      DO 8 J=1,NB2
    8 SS(I)=SS(I)+HH(I,J)*GB(J)
      DO 9 I=1,NB1
      II=I+NB1
      GRADB(I+M)=GRAD(I+M)
      A=SS(I)
      B=SS(II)
    9 S(I+M)=CMPLX(A,B)
      RETURN
      END
      SUBROUTINE READOUT
      COMPLEX V,C
      COMPLEX VAROUT
      COMPLEX GRAF
      INTEGER TYPE
      COMMON /BLOCK3/ TYPE(200),VALUE(200),F(49,151),NOEL(7)
      COMMON /BLOCK5/ IOUT(200),ITEST(200),NOUT(200),ITTEST(200)
      COMMON /BLOCK7/ ALPHA,FUNCT,NIC,FACTOR,EPS,NN,NB
      COMMON /BLOCK8/ V(200),C(200)
      COMMON /BLOCK10/ AMAG(200),APHASE(200),WSTART,WEND,NRES
      COMMON /BLOCK14/ KP(5),W,FREQADD
      COMMON /BLOCK15/ NIT,JOUT,NGRAPH,NALLOUT
      COMMON /BLOCK16/ GRAF(200,5),JGRAPH(5),IGRAPH(5),NPRINT,SCALE,NITT
      COMMON /BLOCK23/ VAROUT(200),VABS(200),VPHASE(200)
      PI=180./3.14159265
      N=NN-1
      DO 7 I=1,JOUT
      K=NOUT(I)
      IF (NOUT(I).LT.NN) GO TO 5
      IF (ITEST(I).EQ.1) GO TO 1
      VAROUT(I)=V(K)
      GO TO 2
1     VAROUT(I)=C(K)/SCALE
2     VABS(I)=SQRT(VAROUT(I)*CONJG(VAROUT(I)))
      IF (REAL(VAROUT(I)).EQ.0.0) GO TO 4
      VPHASE(I)=ATAN(AIMAG(VAROUT(I))/REAL(VAROUT(I)))*PI
      IF (VPHASE(I).LE.0.0.AND.REAL(VAROUT(I)).LT.0.0) GO TO 3
      IF (VPHASE(I).GT.0.0.AND.REAL(VAROUT(I)).LT.0.0) VPHASE(I)=VPHASE(
     1I)-180.0
      GO TO 7
3     VPHASE(I)=VPHASE(I)+180.0
      GO TO 7
4     VPHASE(I)=90.0
      IF (AIMAG(VAROUT(I)).LT.0.0) VPHASE(I)=-VPHASE(I)
      GO TO 7
5     IF (ITEST(I).EQ.0) GO TO 6
      VAROUT(I)=C(K)/SCALE
      GO TO 2
6     VAROUT(I)=V(K)
      GO TO 2
7     IF (VABS(I).EQ.0.0) VPHASE(I)=0.0
      MRES=9H
      IF (NRES.EQ.1) MRES=9HRESONANCE
      PRINT 9, W,MRES
      DO 8 I=1,JOUT
      A=7HVOLTAGE
      IF (ITEST(I).EQ.1) A=7HCURRENT
      PRINT 10, IOUT(I),A,VABS(I),VPHASE(I)
8     CONTINUE
      RETURN
C
C
9     FORMAT (/////,10X,*FREQUENCY = *,E15.5,5X,A9)
10    FORMAT (1H0,10X,*BRANCH*,5X,I5,3X    ,A7,10X,*ABS VALUE = *,E15.5,
     110X,* PHASE = *,E15.5)
```

END

N 56-

```
      SUBROUTINE ALLOUT
      COMPLEX V,C
      COMPLEX A
      COMPLEX GRAF
      COMMON /BLOCK7/ ALPHA,FUNCT,NIC,FACTOR,EPS,NN,NB
      COMMON /BLOCK8/ V(200),C(200)
      COMMON /BLOCK14/ KP(5),W,FREQADD
      COMMON /BLOCK16/ GRAF(200,5),JGRAPH(5),IGRAPH(5),NPRINT,SCALE,NITT
      COMMON /BLOCK23/ A(200)
      N=NN-1
      PRINT 2, W
      DO 1 I=NN,NB
      A(I)=C(I)/SCALE
 1    CONTINUE
      PRINT 3, (V(I),I=1,N)
      PRINT 4, (A(I),I=NN,NB)
      RETURN
C
C
 2    FORMAT (1H0,///,* FREQUENCY = *,E15.5)
 3    FORMAT(1H0,*THESE ARE THE TREE BRANCH VOLTAGES*,//,10(3X,E12.5))
 4    FORMAT (1H0, * THESE ARE THE LINK CURRENTS *,//,10(3X,E12.5))
      END
      SUBROUTINE GRAPH
      COMPLEX E,CONST,V,C,GRAD
      COMPLEX GRAF
      INTEGER ANP
      COMMON /BLOCK1/ ANP(200),NP(200)
      COMMON /BLOCK5/ FR(200),KR(200),NOUT(200),ITTEST(200)
      COMMON /BLOCK7/ ALPHA,FUNCT,NIC,FACTOR,EPS,NN,NB
      COMMON /BLOCK8/ V(200),C(200)
      COMMON /BLOCK10/ AMAG(200),APHASE(200),WSTART,WEND,NRES
      COMMON /BLOCK14/ KP(5),W,FREQADD
      COMMON /BLOCK15/ NIT,JOUT,NGRAPH,NALLOUT
      COMMON /BLOCK16/ GRAF(200,5),JGRAPH(5),IGRAPH(5),NPRINT,SCALE,NITT
      COMMON /BLOCK21/ NST(5)
      N=NN-1
      DO 2 I=1,NGRAPH
      K=NST(I)
      FR(NPRINT)=W
      KR(NPRINT)=0
      IF (NRES.EQ.1) KR(NPRINT)=1
      IF (JGRAPH(I).EQ.0) GO TO 1
      GRAF(NPRINT,I)=C(K)/SCALE
      GO TO 2
    1 GRAF(NPRINT,I)=V(K)
    2 CONTINUE
      IF (NPRINT.LT.200.AND.W.LT.WEND) RETURN
      DO 27 I=1,NGRAPH
      PRINT 28
      A=7HVOLTAGE
      IF (JGRAPH(I).EQ.1) A=7HCURRENT
      IF (ITTEST(I).EQ.1) GO TO 3
      PRINT 29, IGRAPH(I),A
      GO TO 19
    3 PRINT 30, IGRAPH(I),A
      PI=180.0/3.14159265
      DO 7 J=1,NPRINT
      IF (REAL(GRAF(J,I)).EQ.0.0) GO TO 6
      A=ATAN(AIMAG(GRAF(J,I))/REAL(GRAF(J,I)))*PI
      B=REAL(GRAF(J,I))
      IF (A.LT.0.0) GO TO 4
      IF (A.GE.0.0) GO TO 5
    4 IF (B.LT.0.0) A=A+180.
      GRAF(J,I)=CMPLX(A,0.0)
      GO TO 7
    5 IF (B.LT.0.0) A=A-180.
      GRAF(J,I)=CMPLX(A,0.0)
      GO TO 7
    6 A=90.0
      IF (AIMAG(GRAF(J,I)).EQ.0.0) A=0.0
      IF (AIMAG(GRAF(J,I)).LT.0.0) A=-A
      GRAF(J,I)=CMPLX(A,0.0)
    7 CONTINUE
      AMAX=ABS(REAL(GRAF(1,I)))
      DO 8 J=2,NPRINT
      A=ABS(REAL(GRAF(J,I)))
      IF (A.GT.AMAX) AMAX=A
```

```
  8     CONTINUE
        IF (AMAX.EQ.0.0) AMAX=90.0
        DO 9 J=1,NPRINT
        NP(J)=REAL(GRAF(J,I))/AMAX*50.0+51.0
  9     CONTINUE
        DO 10 J=1,101
        ANP(J)=1H.
 10     CONTINUE
        PRINT 31, (ANP(J),J=1,101)
        DO 11 J=1,102
        ANP(J)=1H
 11     CONTINUE
        DO 14 KK=1,NPRINT
        ANP(51)=1H.
        LL=NP(KK)
        ANP(LL)=1H+
        A=REAL(GRAF(KK,I))
        W=FR(KK)
        IF (KR(KK).EQ.1) GO TO 15
 12     PRINT 32, W,A,(ANP(J),J=1,102)
        ANP(LL)=1H
        IF (KR(KK).EQ.1) GO TO 17
 13     CONTINUE
 14     CONTINUE
        GO TO 27
 15     DO 16 II=5,15
        ANP(II)=1HX
 16     CONTINUE
        ANP(LL)=1H+
        IF (ITTEST(I).EQ.0) GO TO 24
        GO TO 12
 17     DO 18 II=5,15
        ANP(II)=1H
 18     CONTINUE
        IF (ITTEST(I).EQ.0) GO TO 25
        GO TO 13
 19     AMAX=GRAF(1,I)*CONJG(GRAF(1,I))
        DO 20 J=2,NPRINT
        A=GRAF(J,I)*CONJG(GRAF(J,I))
        IF (A.GT.AMAX) AMAX=A
 20     CONTINUE
        IF (AMAX.EQ.0.0) AMAX=1.0
        B=SQRT(AMAX)
        DO 21 J=1,NPRINT
        A=SQRT(GRAF(J,I)*CONJG(GRAF(J,I)))
        GRAF(J,I)=CMPLX(A,0.0)
        NP(J)=A*100.0/B+1
 21     CONTINUE
        DO 22 J=1,101
        ANP(J)=1H.
 22     CONTINUE
        PRINT 31, (ANP(J),J=1,101)
        DO 23 J=1,102
```

```
      ANP(J)=1H
23    CONTINUE
      DO 26 KK=1,NPRINT
      ANP(1)=1H.
      LL=NP(KK)
      ANP(LL)=1H+
      A=REAL(GRAF(KK,I))
      W=FR(KK)
      IF (KR(KK).EQ.1) GO TO 15
24    PRINT 32, W,A,(ANP(J),J=1,102)
      ANP(LL)=1H
      IF (KR(KK).EQ.1) GO TO 17
25    CONTINUE
26    CONTINUE
27    CONTINUE
      NGRAPH=201
      RETURN
C
C
28    FORMAT (1H1)
29    FORMAT (1H0,30X,*BRANCH*,I3,5X,    A7     ,6X,*MAGNITUDE PLOT *)
30    FORMAT (1H0,30X,*BRANCH*,I3,5X,  A7      ,6X,*PHASE PLOT *)
31    FORMAT (1H0,1X,*FREQUENCY*,6X,*VALUE*,6X,101A1)
32    FORMAT (1HZ,E11.4,2X,E11.4,3X,102A1)
      END
```

APPENDIX E

General Comments Regarding Use of CANDOFD.

1. Number of nodes cannot exceed 50

2. Number of branches cannot exceed 200

3. Cannot exceed 5 graphical outputs

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

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

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

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

8. CANDOFD can handle zero valued elements as long as there are no loops of short circuits and no cutsets of open circuits. In this context, zero valued R's and L's, and independent voltage sources are short circuits, and zero valued C's and independent current sources are open circuits.

9. If a DC network analysis is to be performed, there can be no independent current source - capacitance cutsets, and no independent voltage source - inductance loops.

10. We cannot have both of the variables FACTOR and FREQADD simultaneously equal to zero.

11. Core requirements:

Since our Fletcher-Powell minimization scheme requires the storing of an N × N matrix, where

$$N = 2 * (NB - \# \text{ of independent sources})$$

it would be inconvenient to always dimension the above matrix 400 × 400.

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

DIMENSION HH(N, N)

where N is the largest number to be encountered in the network (s) to be solved.

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

$$29500 + N^2 \text{ (decimal)}$$

storage locations.

12. In general, one should make use of the automatic current scaling (NSCALE ≠ 1) and of the optimal tree picking (NTREE = 1) algorithms at each frequency point.

13. In general, unless the network is lossless, one need not test for ideal resonances (IDRES = 1) and one need not output the tree and $\underset{\wedge}{F}$ information (NCONT ≠ 1)

14. Comments 12 and 13 will, in general, reduce the execution time required for the analysis of a network. Sample problem #2 indicates variations in CP time caused by different selections of the control variables discussed above.

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

16. Whenever FREQADD ≠ 0.0, CANDOFD assumes that the user wishes to increment the frequency additively.

17. An attempt to graph a signal which is identically zero over the entire frequency range of interest, will result in an arithmetic error during the execution of CANDOFD.

Bibliography:

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

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

(3) R. A. Rohrer, private communication.