NATIONAL RADIO ASTRONOMY OBSERVATORY
CHARLOTTESVILLE, VIRGINIA

ELECTRONICS DIVISION INTERNAL REPORT No. 199

AN ANALYSIS ROUTINE FOR ARBITRARY 2-PORT NETWORKS ON THE H.P. 9845 COMPUTER

DAN L. FENSTERMACHER*

AUGUST 1979

NUMBER OF COPIES: 150

* SUMMER STUDENT.

AN ANALYSIS ROUTINE FOR ARBITRARY 2-PORT NETWORKS ON THE H.P. 9845 COMPUTER

ABSTRACT

The program FARANT was developed as an aid in microwave circuit analysis for the electronics research division at NRAO. It is designed to run on the HP 9845A desktop computer which has graphics and subroutine capabilities, a CRT display and paper printer, cassette storage, and a particularly easy to use syntax and flow of control. It is modelled after BAMP (an HP program for the 9830) and COMPACT (available from TYMSHARE, INC.), and combines strengths from both of these as well as its own capabilities. As of this date, FARANT offers full frequency analysis of arbitrary networks of two-ports, user-specified topology, outputs in various parameter sets for the composite two-ports, and direct plotting capabilities for the scattering parameters. It is also flexible enough to incorporate the user's own BASIC programs for optimization, plotting and many individual analysis problems.

ACKNOWLEDGMENTS

For their guidance and continuing encouragement I cannot pass up the opportunity to thank Sandy Weinreb and John Granlund of the electronics division in Charlottesville. The direction they gave to my efforts was gratefully accepted both for the overall support it provided as well as the numerous occasions where technical consultation became mandatory. Without their help the project could not have come into being.

CONTENTS

USER'S MANUAL FOR FARANT

Introduction

To load the entire program into the HP 9845 memory the user must execute the statement LOAD "FARANT" with the tape in the right-hand reader. He should then press the SPACE DEP key (while holding down the CONTROL key) to set the calculator in "space dependent" mode which basically recognizes multi-character variable names when separated from other words by a blank space.

The program has all of its 3-digit line numbers reserved for the user's commands. The easiest way to enter these is by executing AUTO 100 which will create line numbers automatically. Each line must be STOREd, but can always be changed by executing EDIT LINE n, and then re-entering the line. When the user is satisfied that he has stored his circuit and requested the right outputs, he simply pushes the RUN key, and the analysis is under way.

The program is not interactive, but rather "cooperative". The user types in requests for valid calculations in his section of the program, and when run, the program cooperates. In this sense, FARANT is passive and totally vulnerable to the user's requests or manipulations.

Conventions

The units used throughout the program for input and output are:

Ohms	p Seconds (10^{-12})
m Mhos (10^{-3})	G Hz (10^{9})
p Farads (10^{-12})	Degrees
n Henries (10^{-9})	Inches

Two-port parameters are stored in matrices of dimension 5 x 4 which contain the type of parameters in the (5,1) location (See "From the Program's Point of View..."). For practical purposes, the user need only reference these arrays with a matrix identifier (an asterisk in parentheses) after the letter name, which can be A through H. Thus, the matrix C(*) can hold parameters for a two-port and can then be considered to <u>be</u> that two-port.

In calling some subroutines the type of matrix is required as an input. The five parameter sets that FARANT uses are coded with numbers as follows:

[A] = 1	(ABCD matrix)
[Z] = 2	(impedance matrix)
[Y] = 3	(admittance matrix)
[S] = 4	(scattering matrix)
[T] = 5	(transmission matrix)

Multi-character variables and subroutine names are denoted with one capital letter and the rest small. Once the user has put the computer in "space dependent" mode, however, he need not worry about such details, and can type everything in the normal (capital) letters.

User Commands

Frequency

The user must specify (on his first run) the frequencies for analysis at the beginning of his commands. This can take on a number of forms, but all must assign values (in GHz) to the variable F which is reserved for frequency. Some useful ways of doing this are to put in line 100:

F = 5	- specifying 5 GHz as the only frequency
FOR F = 0 TO 10	- range of integer frequencies
or FOR F = 3 TO 6 STEP .1	- range with specified step size

.

.

.

NEXT F - needed to end the frequency loop

Topology

Within the frequency loop the circuit must be specified as a network of two-ports. This takes the form (noting that the parameter list can be values, literals, variables, or numeric expressions of either sign):

CALL element or function (list of parameters passed to the subroutine)

In most cases the first parameter is a name for the two-port being created, e.g., A(*). The elements are described below and consist of R-L-C circuits, transmission lines, controlled sources, etc. The functions are to Cascade, Series or Parallel two elements, to change ports, to create a branch connection, to print the parameters for an element, and so forth. First, I'll describe the CALLing of the subroutines to perform these operations, and then go through an example.

R-L-C 2-Ports:

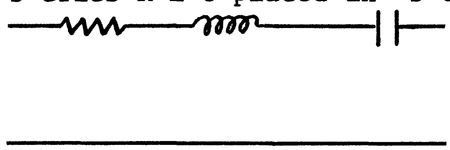
CALL Rlc (X(*),Type$,R,L,C,Place$)

X(*) is an array identifier naming the two-port; a letter A through H followed by (*).

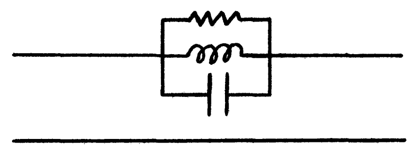
Type$ and Place$ refer to the type of R-L-C (series or parallel) placed in series or parallel in the two port. Thus, both parameters require either "S" or "P", and the quotes are necessary.

R,L,C are the values of resistance, inductance and capacitance in units of ohms, nano-Henries and pico-Farads. A value of zero denotes the lack of that element in the two port (which is sometimes equivalent to assigning it infinite value and sometimes zero).

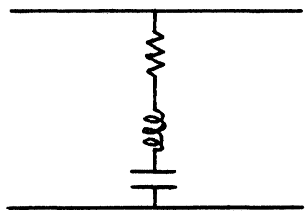
A "S"eries R-L-C placed in "S"eries



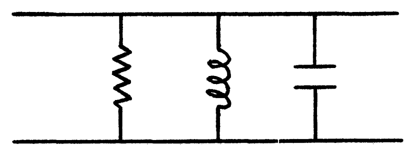
"P" in "S"



"S" in "P"



"P" in "P"



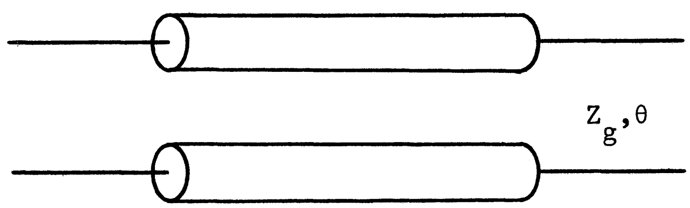
Lossless Transmission Lines (TEM):

CALL Trline(X(*),Zg,Length,K)

Zg is the (real) characteristic impedance of the lossless line $\sqrt{\frac{L}{C}}$.

Length is in inches, the physical dimension.

K is the relative dielectric constant--$\frac{\varepsilon\mu}{\varepsilon_o\mu_o}$ for permeable dielectrics.



Lossy Transmission Lines (TEM):

CALL Lossyline(X(*),Zg,Length,K,Ac,Ad,Fo)

Ac is the attenuation in dB/inch that the line would have with only conductor losses.

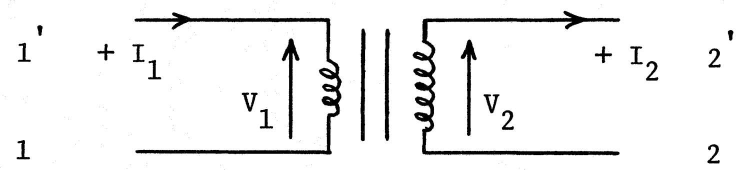
Ad is the same, but for the dielectric losses only.

Fo is the frequency at which Ac and Ad were determined.

Ideal Transformers:

CALL Tf(X(*),Turns1,Turns2)

$\frac{\text{Turns1}}{\text{Turns2}}$ is the actual turns ratio and can be negative to reverse the polarity of port 2's current and voltage. Shown is the positive sense of I and V for a positive turns ratio:



Controlled Sources:

CALL Source(X(*),Control$,Source$,Gain,R1,R2,Delay)

Control$ is either "C" or "V" for a current or voltage controlled source.

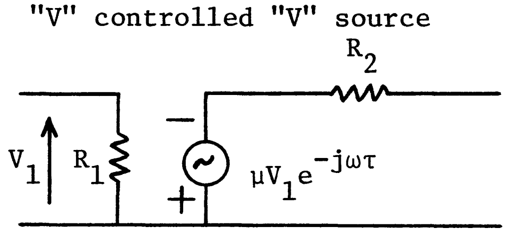
Source$ is also "C" or "V" for the source itself.

Gain is either μ, α, r_m, g_m where the trans-conductance g_m is given in mMhos. The convention is to have positive current into port 2 when shorted, but the Gain can, of course, be negative.

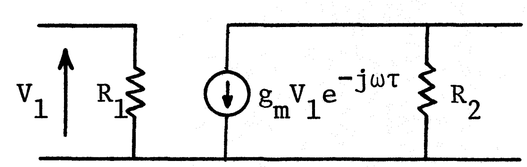
R1,R2 are the resistances at ports 1 and 2 which can be in the range 0 to 10^{20} or so.

Delay is the time lag between the control and the source, in pSec.

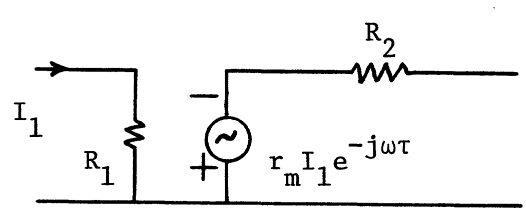
"V" controlled "V" source



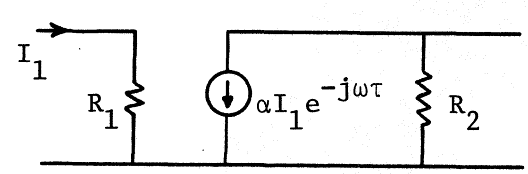
"V" controlled "C"



"C" controlled "V"



"C" controlled "C"



Using Measured Parameters:

CALL Pread(X(*))

In addition to the CALL, the user must store his data in lines 4200 to 4250 with DATA statements. The necessary values are:

Pset (1 to 5), Number of Frequencies, F_1, F_2, F_3 ... F_n ($n \leq 50$),

Data in the form of Magnitude, Phase (degrees) for each parameter 11, 12, 21, 22 at each frequency listed.

The frequencies must be in increasing order. All of these requirements are displayed when EDIT LINE 4200 is executed from the keyboard, thus they can be checked before the data is stored.

For only 1 frequency typed-in the parameters are taken as constant, independent of frequency. For 2, the CALL will perform a linear interpolation, and for 3 or more it will perform a full parabolic interpolation. Frequencies out of the range are extrapolated without warning to the user, and he should use his common sense in this matter.

Exchanging Ports 1 and 2:

CALL FLIP(X(*))

Whatever element is named for X(*), the CALL will swap the ports, thus changing the convention of input and output. (See "Calculations and Logic in the Subroutines" for the conventions).

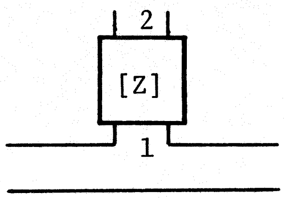
Creating a Branch Element from a 2-Port:

CALL Branch(X(*) ,Type$)

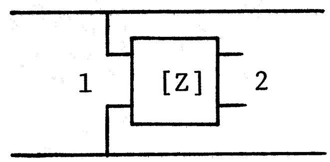
Type$ is either "S" or "P" for a series or parallel branch.

NOTE: Port 1 of X(*) is used to connect the branch to the 2-port, and port 2 is left open and inaccessible.

"S"eries Branch



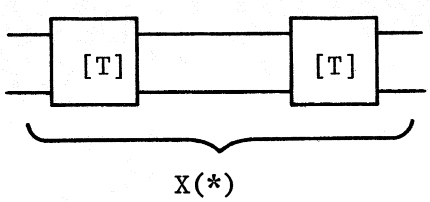
"P"arallel Branch



Cascading 2-Ports:

CALL Cas(X(*),A(*))

X(*) and A(*) can be the same element, in which case a duplicate of X(*) is cascaded onto its right-hand port. X(*) holds the result. A(*) is always cascaded onto the right of X(*).



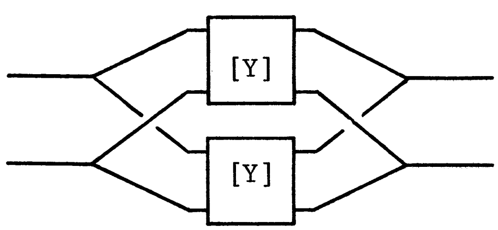
(A(*) is unchanged provided A(*) ≠ X(*))

Paralleling or Putting 2-Ports in Series:

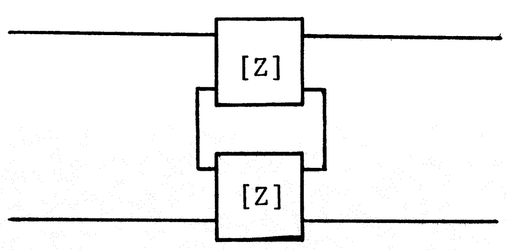
CALL Par(X(*),A(*))

CALL Ser(X(*),A(*))

Again, the result is placed in X(*) and A(*) is unchanged (provided, of course, that A(*) is a different element).



The Par Connection



The Ser Connection

Note that the Ser and Par connections will effectively embed each element between 1:1 transformers (not shown) that, for example, avoid the possibility of shorting the terminals of one 2-port through the other.

Storing and Printing Parameter Sets:

CALL Prt(X(*),Pset)

Pset is 1 to 5 for the various types of parameters. This CALL stores the requested parameters in memory and also prints them. Note that this CALL must be within the user's frequency loop in order to store the information at each frequency (as well as print the parameters). Rollett's stability factor k is also printed, regardless of the type of parameters.

Printing Other Parameters:

CALL Dtrans(Pset)

Once a frequency loop has been completed which included the CALL Prt (Pset) statement, any other parameters can be quickly printed by this CALL statement. Alternatively, if FARANT is run again with this as the sole user instruction, and PRINTER IS 0 is executed just before the run, a hard copy will be produced by the heat sensitive paper printer. This CALL can be repeated to request any of the 5 parameter sets.

Plotting S-Parameters on a Smith Chart:

CALL Smith(Xmin,Xmax,Ymin,Ymax)

CALL Splot(I,J)

Xmin...Ymax are the extreme values of a particular S-Parameter (S_{ij}) that the user wants to plot. (He can find these from the values in the printout at 180°, 0°, -90°, 90°, respectively.) For S_{11} or S_{22}, the largest full Smith chart is obtained with -1, 1, -1, 1, but values smaller than 1 are acceptable as well, to obtain any (blown-up) section of the chart.

I,J are the subscripts of the S parameter to be plotted, e.g., 2,1 is the forward gain.

The graph will be displayed as it is plotted, and when finished, the user can label it by first executing the LETTER statement, and then produce one or more hard copies by pushing STOP and then executing DUMP GRAPHICS.

Graphs cannot be plotted if the CALL Prt was not included in the frequency loop during the analysis. However, subsequent runs of FARANT will retain the data used in plotting, and can skip the entire frequency loop just to plot parameters.

Summary for the User

Insert the tape in right-hand reader.

Execute LOAD "FARANT"

Execute AUTO 100

CALL statements within the frequency loop:

```
Rlc(X(*),Type$,R,L,C,Place$)

Trline(X(*),Zg,Length,K)

Lossyline(X(*),Zg,Length,K,Ac,Ad,Fo)

Tf(X(*),Turns1,Turns2)

Source(X(*),Control$,Source$,Gain,R1,R2,Delay)

Pread(X(*))

Flip(X(*))

Mtrans(X(*),Pset)

Branch(X(*),Type$)

Cas(X(*),A(*))

Par(X(*),A(*))

Ser(X(*),A(*))

Prt(X(*),Pset)
```

CALL statements placed after (and not requiring) the frequency loop:

```
Smith(Xmin,Xmax,Ymin,Ymax)

Splot(I,J)

Dtrans(Pset)
```

Example: Lumped Model of an FET

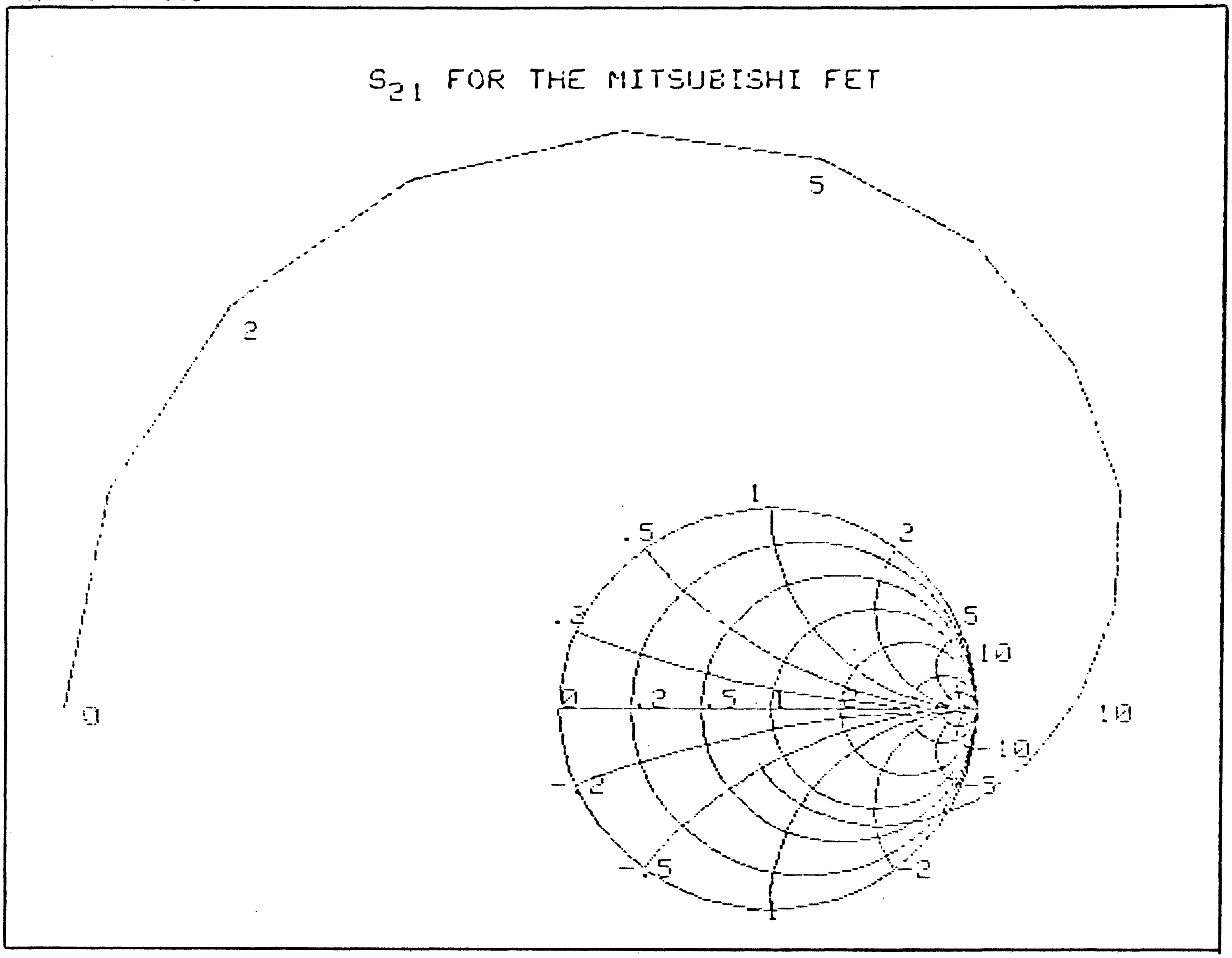
I have executed a LOAD "FARANT", and put the HP9845 in space dependent mode.

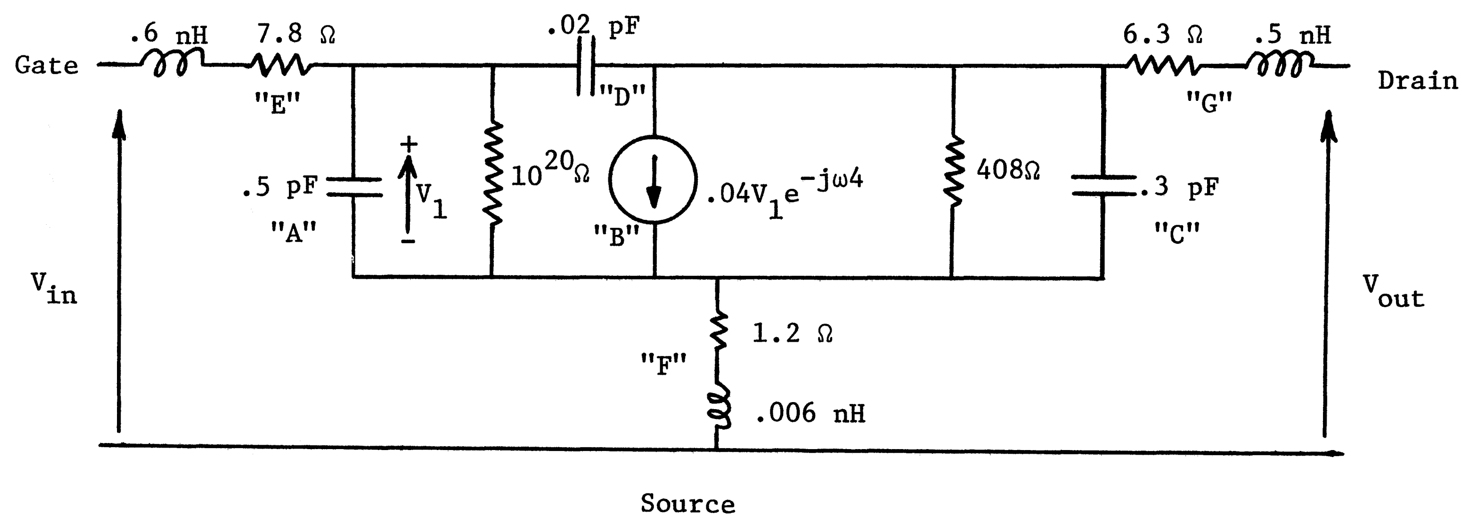
```
AUTO 100

100 FOR F=0 TO 20                 !REQUESTING INTEGER FREQUENCIES FROM D-C TO 20GHz
110   CALL Rlc(A(*),"S",0,0,.5,"P")
120   CALL Source(B(*),"V","C",40,1E20,408,4)
130   CALL Rlc(C(*),"S",0,0,.3,"P")
140   CALL Cas(A(*),B(*))
150   CALL Cas(A(*),C(*))
160   CALL Rlc(D(*),"S",0,0,.02,"S")
170   CALL Par(A(*),D(*))
180   CALL Rlc(E(*),"S",7.8,.6,0,"S")
190   CALL Rlc(F(*),"S",1.2,.006,0,"P")
200   CALL Rlc(G(*),"S",6.3,.5,0,"S")
210   CALL Cas(E(*),A(*))
220   CALL Cas(E(*),G(*))
230      CALL Prt(E(*),4)         !REQUESTING [S] PARAMETERS FOR THE COMPOSITE CKT
240 NEXT F                        !END OF THE FREQUENCY LOOP FOR THIS ANALYSIS
250   CALL Smith(-4,2,-1.2,3.5)   !S21 FOR THE ACTIVE FET HAS GAIN ~3 OR SO
260   CALL Splot(2,1)

EDIT LINE 210
201   CALL SER(A(*),F(*))         !I INSERT THIS LINE NOW IN "EDIT LINE" MODE

PRINTER IS 0

RUN
```

[S] PARAMETERS IN MAGNITUDE AND PHASE

	11		12		21		22		K
FREQ	MAG	ANG	MAG	ANG	MAG	ANG	MAG	ANG	FACT
.000	1.0000	-.0	.0000	90.0	3.3644	-180.0	.7939	-.0	1.00
1.000	.9893	-19.4	.0118	76.0	3.3406	161.0	.7885	-12.4	.14
2.000	.9585	-39.1	.0231	61.9	3.2686	142.0	.7729	-24.9	.28
3.000	.9110	-59.2	.0333	47.9	3.1469	123.0	.7481	-37.6	.44
4.000	.8531	-80.0	.0419	34.0	2.9768	104.1	.7158	-50.8	.61
5.000	.7914	-101.3	.0485	20.4	2.7640	85.4	.6790	-64.4	.80
6.000	.7407	-122.9	.0529	7.3	2.5197	67.3	.6412	-78.6	1.01
7.000	.7021	-144.2	.0551	-5.1	2.2580	49.7	.6061	-93.6	1.26
8.000	.6806	-164.5	.0555	-16.7	1.9931	32.9	.5774	-109.4	1.54
9.000	.6752	176.8	.0543	-27.5	1.7368	17.0	.5580	-125.9	1.95
10.000	.6821	160.3	.0520	-37.4	1.4973	1.9	.5499	-142.8	2.21
11.000	.6972	145.9	.0489	-46.3	1.2796	-12.3	.5536	-159.4	2.60
12.000	.7160	133.5	.0454	-54.2	1.0860	-25.5	.5679	-175.3	3.04
13.000	.7382	122.9	.0417	-61.2	.9171	-37.8	.5903	170.0	3.52
14.000	.7596	113.7	.0381	-67.3	.7720	-49.2	.6179	156.8	4.04
15.000	.7802	105.7	.0346	-72.5	.6487	-59.7	.6478	144.9	4.60
16.000	.7994	98.7	.0314	-76.9	.5449	-69.4	.6781	134.5	5.19
17.000	.8170	92.6	.0285	-80.5	.4583	-78.2	.7073	125.2	5.81
18.000	.8329	87.2	.0259	-83.5	.3862	-86.3	.7345	117.0	6.45
19.000	.8472	82.3	.0237	-85.9	.3263	-93.8	.7595	109.7	7.11
20.000	[illegible]	78.0	.0217	-87.7	.2767	-100.6	.7819	103.2	7.78

LETTER
DUMP GRAPHICS





DETAILED USER INSTRUCTIONS

The Reference Z_o for S-Parameters

The reference or normalization impedance for all S-parameters defaults to 50 ohms when FARANT is run. The user can change this, however, by assigning Z_o in his program segment or editing line 30 to the desired Z_{ref}. The use of transformers also comes in handy for changing the reference of a set of S-parameters measured in some other than 50 ohm system. For instance, after the device parameters are read into a matrix by the subroutine Pread, they can be changed to another reference impedance by sandwiching the 2-port between ideal transformers. The impedance ratio--the square of the turns ratio--must transform the relative impedance from Z_{ref} of the initial parameters (next to the device) to Z_o of the new parameters at the outside ports. Transformers can arbitrarily normalize the input and output to any line impedance that is desired.

Perhaps a simpler way of changing the Z_{ref}, however, is to use the subroutine Mtrans to change from [S] parameters to either [A], [Y] or [Z], with Z_o temporarily set equal to Z_{ref} of the initial device parameters. The variable Z_o is used by Mtrans as a common storage location for the normalization impedance. Thus, the statements CALL Pread(A(*)), Z_o = 73, CALL Mtrans(A(*),2), Z_o = 50 would change the 73 ohm S-parameters in A(*) to [Z] parameters and maintain a consistent 50 ohm system.

Creating Storage for More Elements

In line 15 the program sets up storage matrices for 8 elements named A through H. There is nothing special about the number 8 or those particular letters, and the user should feel free to add to the list or change the names according to his desires. Each element dimensioned takes about 170 bytes of memory, whether or not is is used in later statements. Element names in lines 100 - 999 must, of course, correspond to those dimensioned in earlier program lines.

The data-base matrix Dat(101,9) is not so flexible, however. Both Dtrans and Splot contain a loop of 100 iterations to step through the data-base. Several other places assign the single element Dat(101,1) the type of parameters in the data-base. Thus, the user can expect trouble if he tampers with the data-base.

Special Frequency Loops and the Data Base

It is possible that the user would like analysis over a specialized set of frequencies not specified by a simple FOR-NEXT loop. Frequency bands, for example, can be specified by a loop construction as follows:

```
FOR I = 1 to 2                  - two bands
READ F1, F2, DeltaF             - initial F, final F, and increment
DATA 0, 10, 1, 15, 100, 5
FOR F = F1 to F2 STEP DeltaF    -  0 to 10 GHz by 1
    .                           - 15 to 100 GHz by 5
    .
    .
NEXT F
NEXT I
```

Using the READ-DATA statements in combination with FOR-NEXT loops, one can tailor any frequency specification he desires.

It is also a simple matter to add frequencies to a printout and the database. All that must be done is to delete line 25 which sets Count = 0. When the program is then RUN, it will not know that the analysis has started again (because count is held in common between runs), and it will continue the printout and writing into the data base as if nothing happened.

There are a few peculiarities about this procedure, however, that can cause mysterious results if they are not understood. First of all, the data base does not get erased upon RUNning FARANT, specifically for this purpose. To begin with a clean slate then, it is advisable to execute SCRATCH C to initialize the common variables to zero again when that is desired. Since the entire data base is one entity, it should only contain one type of parameters, and since plotting uses the entire data base for each graph, the frequencies should all be increasing if one is planning to do plots.

Saving the Circuit Topology or Measured Parameters

Oftentimes there is a considerable amount of work involved in typing in a circuit or the measured parameters for a device. When more work must be done on the same topology, it can all be stored very simply by executing the statement STORE "circuit name (≦ 6 characters)". The corresponding statement on later runs is then LOAD "circuit name". This will take less than 30 seconds each way, but is wasteful of tape storage and inflexible.

Greater efficiency of data handling (although slower by word) is accomplished by the SAVE-GET statements. SAVE "FETDAT",4200,9999 would save the last part of FARANT, including the data, and SAVE "CKT",1,999 would save the first part. Then the entire program could be compiled into memory by the statements GET "CKT", GET "FARSUB",1000 and GET "FETDAT",4200 where "FARSUB" is a SAVEd version of FARANT's subroutines. These pieces are flexible enough then to allow many topologies or device parameters to be re-used from tape storage. It should be understood that the GET statement writes the program lines into memory beginning at the specified line number, uses the line increments that were saved, and deletes all higher numbered lines in memory.

Alternate Plotting and Analysis

Because FARANT is designed to allow the user complete control, it will accept virtually any Basic statements or programs in the user's area of the program. These can range from the simplest calculations or matrix operations to plotting routines or entire programs, any of which can be linked into memory with the GET "   ",n statement.

As an example, say one wanted a plot of the magnitude of Z_{22} vs. frequency over a previously analyzed frequency range. In general this is what he would do:

```
CALL Dtrans(2)                    - to get [Z] if he hadn't already
                                    printed them

GRAPHICS

GCLEAR

SCALE 0, 20, 0, 100               - 20 GHz, max output impedance of
                                    100 ohms

AXES

MOVE Dat(1,1),SQR(Dat(1,8)^2 +
     Dat(1,9)^2)

FOR I = 1 to Count

    PLOT Dat(I,1),SQR(Dat(I,8)^2 +
         Dat(I,9)^2)

    NEXT I
```

Since the frequency and parameters (in any form) are held in the data base, plots are as easy as this.

Other analysis problems can similarly be incorporated into the user's commands. For instance, he could set up a loop to vary a parameter value, or an iterative process that optimized some aspect of the circuit performance numerically. Noise figure calculations and optimizations are in fact two features yet to be incorporated into the subroutines of FARANT. Any adventuresome programmer is more than welcome to undertake these tasks and thereby widen the applicability of the program for others.

FROM THE PROGRAM'S POINT OF VIEW...

Complex Number Manipulations

The circuit analysis that FARANT uses requires complex numbers in most calculations. Voltages and currents are considered to be "phasors" of some magnitude and phase, with rotation at the frequency ω assumed. Impedances, admittances, and gains with delays are all taken to be complex numbers and are manipulated with both real and imaginary parts. The Euler formula for expansion of EXP(imaginary number) is used extensively:

$$e^{j\theta} = \cos\theta + j\sin\theta$$

For example, a transmission line which takes τ seconds to propagate an E-M wave has a phase delay of voltage and current (at F Hz) of $2\pi F\tau$ radians, or an $e^{-j(2\pi F)\tau}$ phase factor.

Alternatively, given a complex number, it can be interpreted as a phased quantity by a magnitude and angle:

$$a + jb = \sqrt{a^2 + b^2} \text{ at angle } \tan^{-1}\left(\frac{b}{a}\right) + (\pi \text{ if } a < 0)$$

Complex algebra can effectively be handled with matrix algebra by using the following mapping:

$$z_1 = a + jb \leftrightarrow \begin{bmatrix} a & b \\ -b & a \end{bmatrix}$$

$$\text{Mag } (a + jb) \leftrightarrow \left\{ \text{Det} \begin{bmatrix} a & b \\ -b & a \end{bmatrix} \right\}^{\frac{1}{2}}$$

$$z_1 \; z_2 \leftrightarrow \text{matrix product (order not important)}$$

$$\frac{1}{z_1} \leftrightarrow \begin{bmatrix} a & b \\ -b & a \end{bmatrix}^{-1}$$

2 X 2 matrix of complex z's $\leftrightarrow$ 4 X 4 (partitioned) matrix of real numbers

Product of complex 2 X 2's $\leftrightarrow$ product of corresponding 4 X 4's

To eliminate the ambiguity in the angle of complex numbers, I found that the simplest mapping from a + jb to θ (which avoids division by 0) is the following:

IF a*b = 0 THEN θ = [SGN(a)(SGN(a)-1) + SGN(b)]*90°

IF a*b <> 0 THEN θ = TAN^{-1}(b/a) - [SGN(a)-1]*SGN(b)*90°

(The SGN( ) function yields -1, 0, or 1.)

This maps the arc-tangent function into the range $-180° < \theta \leq 180°$.

Parameter Transformations

SUB Mtrans(X(*),N)

X(5,1) holds the present type of parameters

N is the new form of parameters

$$[Z] = \frac{1}{C}\begin{bmatrix} A & AD-BC \\ 1 & D \end{bmatrix}$$

$$[A] = \frac{1}{Z_{21}}\begin{bmatrix} Z_{11} & Z_{11}Z_{22}-Z_{12}Z_{21} \\ 1 & Z_{22} \end{bmatrix}$$

$$[Y] = [Z]^{-1}$$

$$[Z] = [Y]^{-1}$$

$$[S] = \Big[[1] - [Y]Z_0\Big] \times \Big[[1] + [Y]Z_0\Big]^{-1}$$

$$[Y] = \Big[[1] - [S]\Big] \times \Big[[1] + [S]\Big]^{-1} \frac{1}{Z_0}$$

$$[T] = \frac{1}{S_{21}}\begin{bmatrix} 1 & -S_{22} \\ S_{11} & S_{12}S_{21}-S_{11}S_{22} \end{bmatrix}$$

$$[S] = \frac{1}{T_{11}}\begin{bmatrix} T_{21} & T_{11}T_{22}-T_{12}T_{21} \\ 1 & -T_{12} \end{bmatrix}$$

$$[A] = \frac{1}{2}\begin{bmatrix} T_{11} + T_{12} + T_{21} + T_{22} & Z_0\,(T_{11} - T_{12} + T_{21} - T_{22}) \\ \frac{1}{Z_0}\,(T_{11} + T_{12} - T_{21} - T_{22}) & T_{11} - T_{12} - T_{21} + T_{22} \end{bmatrix}$$

$$[T] = \frac{1}{2}\begin{bmatrix} A + \frac{B}{Z_0} + CZ_0 + D & A - \frac{B}{Z_0} + CZ_0 - D \\ A + \frac{B}{Z_0} - CZ_0 - D & A - \frac{B}{Z_0} - CZ_0 + D \end{bmatrix}$$

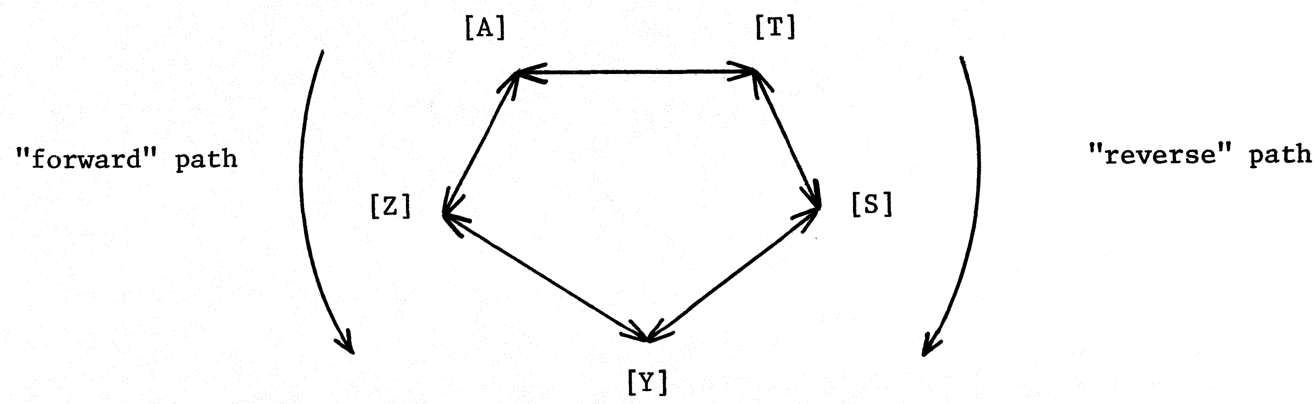
Note convention:

[A] = 1   [S] = 4

[Z] = 2   [T] = 5

[Y] = 3

Mtrans can perform the following cyclical transformation:



These were chosen on the basis of their simplicity (for example, 1 to 2 and 2 to 1 are equivalent transformations) and their completeness in closing the loop. The speed in stepping through the loop is minimal because the algorithm takes the shortest path (if possible) and usually does only 1 or 2 transformations.

Any matrix type may be desired as a termination point:

[Z]	- for SERies connections
[Y]	- for PARalleling
[A] or [T]	- for CAScading (both work)
[S]	- for output and plotting

<u>Handling undefined matrices</u> (ones that are infinite):

In many cases, a particular matrix will not be defined because it is unbounded. For example, the [Y] matrix of a parallel impedance, [Z] of a series impedance, [Y] or [Z] of a lossless half-wave transmission line, [A] or [T] of a "broken" or "shorted" 2-port, and even [S] for a few special cases can be infinite in the ideal case. Given a defined matrix and a transformation to be performed, however, one can always tell if the result is finite by checking for division by 0 or the inversion of a matrix whose determinant is 0.

Since there exist two possible "paths" to the desired matrix, this cyclical transformation will only fail if:

1. the final matrix is infinite
2. [S] <u>and</u> [Z] don't exist, or
3. [A] (and [T]) and [Y] don't exist (note that [A] and [T] are finite or infinite together)

Now case 2 is actually impossible, and we assume case 3 can't happen for any meaningful 2-port. Thus, the only consideration is what to do when the <u>requested</u> matrix is infinite. Each subroutine that uses Mtrans deals with that locally.

<u>Flow of Control in Mtrans:</u>

Program Line	Matrix Change (P→N)
1_1	1 → 2
1_2	2 → 3
1_3	3 → 4
1_4	4 → 5
1_5	5 → 1
GOTO 1_1 (forward path)	
- - - - - - - - - -	
1_6	5 → 4
1_7	4 → 3
1_8	3 → 2
1_9	2 → 1
1_{10}	1 → 5
GOTO 1_6 (backward path)	

Mtrans first chooses the shortest path by the statements:

IF $\overbrace{(N>=P)*(N-P) + (N<P)*(5+N-P)}^{\text{forward distance}}$ <3 THEN

ON P GOTO $1_1, 1_2, 1_3, 1_4, 1_5$

ON 6-P GOTO $1_6, 1_7, 1_8, 1_9, 1_{10}$

(P = present type of parameter set)

If a infinite matrix is encountered, the variable Nogo is set equal to 1. This variable is held in common so all other subroutines can check Nogo after calling Mtrans to see if it failed. In all cases X(5,1) gets the type parameters of the final matrix that Mtrans obtained.

At each 1_i the following occurs:

1. If N = [matrix type to be changed], then the changes have been successful; set Nogo = 0 and finish.
2. If change is impossible AND Nogo = 1 already, then set X(5,1) = P and subexit. (Failure has occurred in both directions.)
3. If change is impossible but Nogo is still 0, then set Nogo = 1 and branch to the other path to try that.
4. Perform the change normally and then set P = [the type parameters just created].

Initializations in the Main Program Segment

OPTION BASE 1

This sets the default lower-limit of dimensioned arrays to 1 rather than 0. Thus, DIM A(5,4) creates an array whose rows are 1 to 5 and columns 1 to 4.

This statement is needed in every subroutine as well.

COM Nogo,Zo,F,Count,SHORT Dat(101,9)

Here common storage space is set up for these variables so that all subroutines can use them, and they remain assigned even when the user RUNs his program again (unless they are explicitly re-assigned).

Nogo - flag for inability to get some requested parameters (see Mtrans)

Zo - reference impedance for [S] parameters; set = 50

F - frequency; the user must call his frequencies with the name F for the subroutines to work

Count - number of rows of Dat(101,9) containing stored frequency data; set = 0

SHORT Dat(101,9) - This is the data base holding up to 100 frequencies and 8 parameter values (real, imaginary) for each. Dat(101,1) holds the type of parameters. SHORT precision allows 6 significant digits to be stored with an exponent between -63 and +63. Dat(*) remains assigned in memory until SCRATCH C is executed which initializes it to 0.

DIM A(5,4),B(5,4),C(5,4),...H(5,4)

This sets up 8 matrices to hold parameters for different circuit elements. The size must be 5 x 4 because element (5,1) is needed to hold the type of parameters, and the 4 x 4 square is a partitioned 2 x 2 of complex parameters (see Complex Number Manipulations). Note that the computer distinguishes between F (frequency) and F(5,4) (a matrix named F).

Calculations and Logic in the Subroutines

SUB Rlc(X(*),Type$,R,L,C,Place$)

	Series Type	Parallel Type
Placed in Series	$[A] = \begin{bmatrix} 1 & Z \\ 0 & 1 \end{bmatrix}$	$[A] = \begin{bmatrix} 1 & 1/Y \\ 0 & 1 \end{bmatrix}$
Placed in Parallel	$[A] = \begin{bmatrix} 1 & 0 \\ 1/Z & 1 \end{bmatrix}$	$[A] = \begin{bmatrix} 1 & 0 \\ Y & 1 \end{bmatrix}$
	$Z = R+j(\omega L-1/\omega C)$	$Y = 1/R+j(\omega C-1/\omega L)$

At D-c (0 Hz) the subroutine changes F to 10^{-20} GHz to avoid an open circuit capacitor or shorted inductor. At a frequency where L and C are exactly resonant and R = 0, the logic adds 10^{-10} ohms of resistance to a series L - C in parallel and 10^{-10} ohms of conductance to a parallel L - C placed in series. A value of zero passed to the subroutine causes it to omit the corresponding term in the calculation of Z or Y.

SUB Trline(X(*),Zg,Length,K)

Normalized to a system of Zg ohms,

$$[S] = \begin{bmatrix} o & e^{-j\omega\tau} \\ e^{-j\omega\tau} & o \end{bmatrix}$$

To obtain the parameters in a Z_o system, the subroutine "de-normalizes" using Z_g like so:

$$[Y] = \Big[[1] - [S]\Big] \times \Big[[1] + [S]\Big]^{-1} (1/Z_g)$$

Before taking the inverse though, Trline checks for the DET = 0 indicating a line of length some multiple of $\lambda/2$. In that case 10^{-10} loss is assumed in both s_{12} and s_{21} before de-normalizing the parameters. This effectively adds a "pad" that attenuates the wave by $(1-10^{-10})$ without affecting Z_g or the phase length of the line.

SUB Lossyline(X(*),Zgo,Length,K,Ac,Ad,Fo)

First the conversion is made from dB's/inch to nepers/inch by the equality ln 10 nepers = 20 dB. Then an analysis of John Granlund (Memorandum of 7/31/79 titled "Lossy Transmission Lines") is used to determine the distributed impedance of the conductor and admittance of the dielectric. The formulas I have used represent a theoretical analysis based on good conductors and dielectrics at microwave frequencies. The distributed values John obtained are:

$$L = Z_{go}\sqrt{K}/\underline{c} \quad (\underline{c} = \text{speed of light in free space})$$

$$C = \sqrt{K}/\underline{c}Z_{go}$$

$$R = \sqrt{F/F_o} \;\; 2Ac\left(Ac + \sqrt{2Ac^2 + \omega_0^2 LC}\right) \;\; /\omega_0 C$$

$$G = (F/F_0)\left[2Ad\sqrt{Ad^2 + \omega_0^2\, LC}\right]/\omega_0 L$$

From these the series impedance per length in the conductor and the shunt conductance per length in the dielectric are:

$$z = R + j(\omega L + R)$$

$$y = G + j(\omega C)$$

The complex characteristic impedance of the line and the propagation constant are directly related to z and y by:

$$Z_g = \sqrt{z/y} \qquad \gamma = \sqrt{zy}$$

The angle of γ is taken between 0 and 90° (not including 0); the angle of Z_g between -45° and 45°. At this point all that is left to calculate are the S-parameters which are the same as in the lossless case except that $\gamma\cdot$Length replaces $\omega\tau$ in the exponential, and the denormalization impedance is complex.

SUB Source(X(*),Control$,Source$,Gain,R1,R2,Delay)

	V Source	C Source
V Controlled	$[Z] = \begin{bmatrix} R_1 & 0 \\ -R_1\mu & R_2 \end{bmatrix}$	$[Z] = \begin{bmatrix} R_1 & 0 \\ -R_1R_2g_m & R_2 \end{bmatrix}$
C Controlled	$[Z] = \begin{bmatrix} R_1 & 0 \\ -r_m & R_2 \end{bmatrix}$	$[Z] = \begin{bmatrix} R_1 & 0 \\ -\alpha R_2 & R_2 \end{bmatrix}$

A non-zero value of Delay changes the Gain term (Z_{21}) into a complex number with a phase factor of $e^{-j2\pi F\tau}$, where τ is the Delay. If either R_1 or R_2 are zero, the subroutine changes them to 10^{-10} ohms to keep [Z] well-behaved.

SUB Pread(X(*))

Any parabolic function can be uniquely specified by 3 constants whose geometric interpretations are the x, y coordinates of the focus and distance to the line by which the parabola is defined. Algebraically, these are represented as a, b, c in

$$p = a + bf + cf^2$$

where "p" is the real or imaginary part of some parameter, and f is the frequency.

If we know the parameters for each of 3 different frequencies, we have a full-rank matrix in the following:

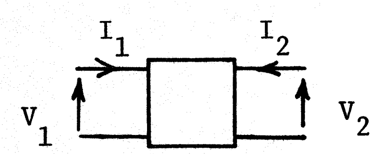
$$\begin{bmatrix} 1 & f_1 & f_1^2 \\ 1 & f_2 & f_2^2 \\ 1 & f_3 & f_3^2 \end{bmatrix} \times \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix}$$

The constants a, b, c are found from the matrix solution of these equations. Whenever possible, the subroutine uses 2 frequencies below F and one above for the interpolation.

SUB Flip(X(*))

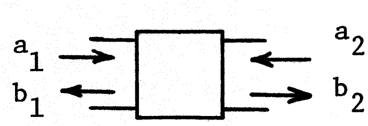
The standard conventions for current and voltage are pictured along with the required matrix operations to exchange ports 1 and 2 for the various parameter sets:

[ABCD] Parameters
$$\begin{bmatrix} V_1 \\ I_1 \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix}_{12} \begin{bmatrix} V_2 \\ -I_2 \end{bmatrix}$$



$$[A]_{21} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad [A]^{-1}_{12} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

[T] Parameters
$$\begin{bmatrix} a_1 \\ b_1 \end{bmatrix} = [T]_{12} \begin{bmatrix} b_2 \\ a_2 \end{bmatrix}$$



$$[T]_{21} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad [T]^{-1}_{12} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

[S] Parameters
$$\begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = [S]_{12} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}$$

[Y] Parameters $\begin{bmatrix} I_1 \\ I_2 \end{bmatrix} = [Y]_{12} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix}$

[Z] Parameters $\begin{bmatrix} V_1 \\ V_2 \end{bmatrix} = [Z]_{12} \begin{bmatrix} I_1 \\ I_2 \end{bmatrix}$

$$[X]_{21} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad [X]_{12} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad \text{for } [X] = [S],[Y],[Z]$$

SUB Branch(X(*),Type$)

Since the open-circuit [Z] parameters are used to define the branch element's input impedance, this subroutine must deal with the possibility of an unbounded [Z] matrix. When this happens, the parallel branch becomes a null element, but the logic assigns $Z_{11} = 10^{10}$ for the series branch. Similarly, for a parallel branch whose Z_{11} is exactly 0, the subroutine uses 10^{10} for its input admittance. The [A] matrix is then assigned as it would be for an R-L-C of known series impedance or shunt admittance. Note that port 2 is left open and is then "inaccessible". If a shorted stub is desired, then it should be cascaded on before the element is made into a branch.

SUB Cas(X(*),A(*))

The subroutine chooses the "closer" of [A] or [T] parameters to use in the cascading process and then multiplies the matrices for each element. If the transmission parameters don't exist (i.e. $\rightarrow \infty$), then there is essentially no "front-talk" occurring. The circuit has effectively been broken. When this happens, the subprogram temporarily goes to [S] parameters and assigns $s_{21} = 10^{-10}$ to allow some front talk. A message is printed, telling the user that this has been done (meaning that an exact analysis at that F would have 2 separate circuits), and the cascading is then performed with the now finite transmission parameters. (If for some fantastic reason the [S] parameters don't exist either, a message is printed that the analysis will be faulty at that frequency.)

SUB Par(X(*),A(*))

The new matrix is placed in X(*) and is simply the sum of the [Y] matrices for each element. If the subroutine is somehow asked to parallel an element with infinite (non-attainable) [Y] parameters, then the matrix that is placed in X(*) is the matrix for that element, in some other parameters. Thus, X(*) is unchanged by the Parallel subroutine if it has infinite [Y] parameters (whether or not A(*)'s [Y] parameters are finite).

SUB Ser(X(*),A(*))

The new matrix placed in X(*) is the sum of the [Z] matrices for each element. The logic for dealing with infinite [Z] parameters is identical for the Par subroutine -- X(*) gets the matrix (in other than [Z] form) for the infinite [Z] element.

SUB Prt(X(*),Pset)

X(*) is the 5 x 4 matrix of any type of parameter for some composite 2-port of interest. When Prt is called, it changes the parameters to type Pset and then supplies a formatted output to the standard printer in magnitude and phase (see Complex Number Manipulations). Since the CALL is placed within the user's frequency loop, the parameters are looked at sequentially and so it is here that FARANT chooses to store the information. By incrementing the Count variable each time it is called, Prt writes the parameters into the common data base Dat(101,9). From there the data can be handled efficiently by other subroutines (see Dtrans,Splot). The heading is only printed before the first frequency (when Count = 1) and this signals the over-writing of the data base as well.

If the requested parameters cannot be obtained at some frequency, then this is printed instead of the parameters, and the data-base gets zeroes for each parameter at that F. The data is, unfortunately, lost at that frequency.

The formatting was designed to accomodate a large variety of circuits, but in some cases (e.g., Z_{11} for a capacitive input near D-C) the values are just too large to fit. Instead of truncating them, however, the 9845 will print them in full glory on a separate line. This destroys the appearance of the format, but does display all the information.

The K factor is calculated in [Y] parameters by the formula:

$$K = \frac{2\ \mathrm{Re}\left[Y_{11}\right]\ \mathrm{Re}\left[Y_{22}\right] - \mathrm{Re}\left[Y_{12}Y_{21}\right]}{\left|Y_{12}Y_{21}\right|}$$

SUB Dtrans(Pset)

Because Dat(101,9) is stored in SHORT precision (6 significant digits), some variation in the rightmost digits can be expected if Dtrans is called several times in succession, each time changing the parameter type. The K factor seems to be noticeably affected at times by this loss of precision.

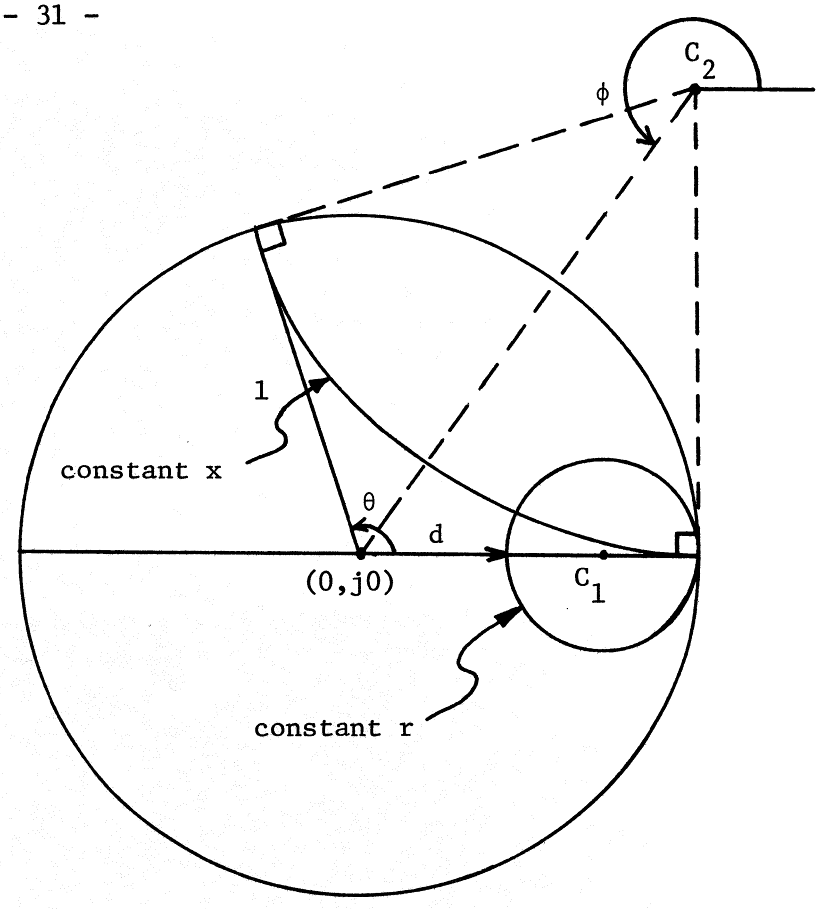
SUB Smith (Xmin,Xmax,Ymin,Ymax)

The circles of constant r and arcs of constant x are plotted incrementally along their length. The geometry is described below:

$$d = \frac{(r + j0) - 1}{(r + j0) + 1} = \left|\Gamma(r)\right|_{x=0}$$

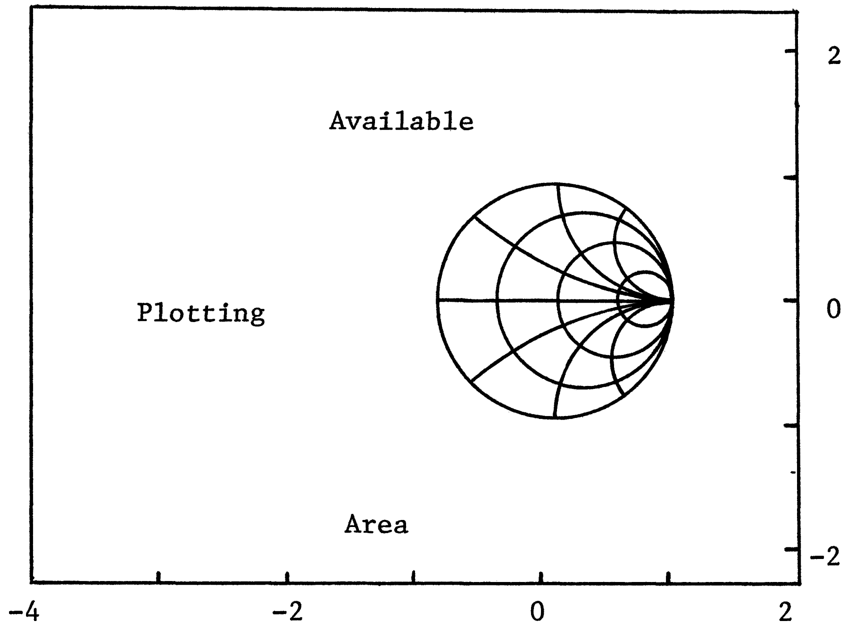
Thus, the center is $C_1\left[\frac{r}{r+1}, 0\right]$

and the radius is $\frac{1}{r+1}$.



From the dotted line construction, it is clear that the center is at $C_2(1, \tan^{-1}\frac{\theta}{2})$ where θ runs from -180° to +180° for the sign to be correct. The radius (height above horizontal line) is $\tan^{-1}\frac{\theta}{2}$. The angular range is simply (180° - θ). For +x values ϕ is from (90° + θ) to 270°; for -x values 90° to (270° - θ).

The size of the chart is determined from the Xmin, Xmax, Ymin, Ymax parameters. The graphics ROM will automatically fit the largest plotting area with those limits into its (rectangular) plotting screen, keeping the horizontal and vertical scale factors the same. Thus, -4, 2, -1, 1 will result in a plotting area as shown:



Values -1, 1, -1, 1 produce the largest full chart. Each call of Smith erases all previous graphics and puts the computer in graphics mode to show the subsequent plotting.

SUB Splot(I,J)

All S parameters can be represented in real,imaginary form as a + jb. Because the Smith Chart can be viewed as Real$\{\Gamma\}$ on the horizontal and Imaginary$\{\Gamma\}$ on the vertical axis, it is convenient to plot all S parameters as if they were reflection coefficients.

Plotting takes place from the internal data-base Dat(101,9) which holds frequency and the four parameters in real,imaginary form for up to 100 frequencies. Splot will automatically change the data-base to S parameters if it is not in that form already.

A useful feature of plotting this way is that the data-base is retained as long as the computer remains on, provided SCRATCH A and SCRATCH C are not executed. Thus one can do a separate run to plot each of the four S parameters without ever recalculating them.

LISTING OF THE PROGRAM

```
5     OPTION BASE 1
10    COM Nogo,Zo,F,Count,SHORT Dat(101,9)   ![DAT] HOLDS FREQ AND DATA IN RE,IMAG
15    DIM A(5,4),B(5,4),C(5,4),D(5,4),E(5,4),F(5,4),G(5,4),H(5,4)
20    FIXED 3
25    Count=0                              !#FREQS CURRENTLY STORED IN DATA BASE
30    Zo=50
35    REM USER SHOULD BEGIN HIS PROGRAM AT LINE 100; USING 3 DIGIT LINE NUMBERS
55    REM ####################################################################################
999   END


1000 SUB Mtrans(X(*),N)                   !CHANGES MAT [X] FROM PRESENT FORM TO New
1005 OPTION BASE 1
1010 COM Nogo,Zo
1015 P=X(5,1)                             !Present TYPE OF PARAMS
1020 REDIM X(4,4)                         !IF P IS X(5,1), IT IS NOT LOST HERE
1025 DIM A(2,2),B(2,2),C(2,2),D(2,2),G(4,4),H(4,4)
1030 Nogo=0
1035 IF (N>=P)*(N-P)+(N<P)*(5+N-P)<3 THEN ON P GOTO 1045,1085,1125,1200.1330
                                          !ENTRANCES TO FORWARD PATH
1040 ON 6-P GOTO 1395,1525,1590,1630,1670      !BACKWARD PATH ENTRANCES
1045       IF N=1 THEN Finish             ![X] IS IN New FORM
1050 GOSUB Partitionx
1055 IF DET(C)<>0 THEN 1075
1060    IF Nogo THEN Failexit             !OTHER ROUTE MUST HAVE FAILED TOO; GIVE UP
1065    Nogo=1                            !BLOCKED GOING SHORT WAY AROUND
1070    GOTO 1670                         !TAKE OTHER (BACKWARD) PATH
1075 GOSUB Azchange                       ![A]1 CHANGED TO [Z]2
1080 P=2                                  !P IS CURRENT PARAMETER TYPE
1085       IF N=2 THEN Finish
1090 IF DET(X)<>0 THEN 1110
1095    IF Nogo THEN Failexit
1100    Nogo=1
1105    GOTO 1630
1110 MAT X=INV(X)
1115 GOSUB Avgdiags
1120 P=3                                  ![X] IS NOW [Y]3 PARAMETERS
1125       IF N=3 THEN Finish
1130 MAT X=X*(Zo)
1135 MAT G=IDN(4,4)
1140 MAT H=G-X
1145 MAT G=G+X
1150 IF DET(G)<>0 THEN 1180
1155    IF Nogo=0 THEN 1170
1160       MAT X=X*(Zo)
1165       GOTO Failexit
1170    Nogo=1
1175    GOTO 1590
1180 MAT G=INV(G)
1185 MAT X=G*H
1190 GOSUB Avgdiags
1195 P=4                                  ![X] IS NOW [S]4
1200       IF N=4 THEN Finish
1205 GOSUB Partitionx
1210 IF DET(C)<>0 THEN 1230
1215    IF Nogo THEN Failexit
1220    Nogo=1
1225    GOTO 1525
1230 MAT C=B-C
```

```
1235 MAT C=INV(C)
1240 MAT H=C
1245 R=C=1
1250 GOSUB Fillx
1255 MAT H=A*C
1260 R=3
1265 C=1
1270 GOSUB Fillx
1275 MAT D=D*(-1)
1280 MAT H=C*D
1285 R=1
1290 C=3
1295 GOSUB Fillx
1300 MAT H=A*D
1305 MAT G=G+H
1310 MAT H=C*G
1315 R=C=3
1320 GOSUB Fillx
1325 P=5                                  ![X] IS NOW [T]5
1330        IF N=5 THEN Finish
1335 GOSUB Partitionx                     ![T] ALWAYS CAN GO TO [A]
1340 FOR R=1 TO 2
1345 FOR C=1 TO 2
1350    X(R,C)=A(R,C)+B(R,C)+C(R,C)+D(R,C)
1355    X(R,C+2)=Zo*(A(R,C)-B(R,C)+C(R,C)-D(R,C))
1360    X(R+2,C)=(A(R,C)+B(R,C)-C(R,C)-D(R,C))/Zo
1365    X(R+2,C+2)=A(R,C)-B(R,C)-C(R,C)+D(R,C)
1370 NEXT C
1375 NEXT R
1380 MAT X=X/(2)
1385 P=1                                  ![X] IS NOW [A]1
1390 GOTO 1045                            !GO ON AROUND FORWARD PATH
1395        IF N=5 THEN Finish            !FROM HERE ON IS THE BACKWARD PATH
1400 GOSUB Partitionx
1405 IF DET(A)<>0 THEN 1425
1410    IF Nogo THEN Failexit
1415    Nogo=1
1420    GOTO 1330
1425 MAT G=A*D
1430 MAT A=INV(A)
1435 MAT H=A*C
1440 R=C=1
1445 GOSUB Fillx
1450 MAT B=B*(-1)
1455 MAT H=A*B
1460 R=C=3
1465 GOSUB Fillx
1470 MAT H=A
1475 R=3
1480 C=1
1485 GOSUB Fillx
1490 MAT H=B*C
1495 MAT G=H+G
1500 MAT H=G*A
1505 R=1
1510 C=3
1515 GOSUB Fillx
1520 P=4                                  ![X] IS NOW [S]4
1525        IF N=4 THEN Finish
1530 MAT G=IDN(4,4)
1535 MAT H=G-X
1540 MAT G=G+X
1545 IF DET(G)<>0 THEN 1565
1550    IF Nogo THEN Failexit
1555    Nogo=1
1560    GOTO 1200
```

```
1565 MAT G=INV(G)
1570 MAT X=G+H
1575 MAT X=X/(Zo)
1580 GOSUB Avgdiags
1585 P=3                                 ![X] IS NOW [Y]3
1590         IF N=3 THEN Finish
1595 IF DET(X)<>0 THEN 1615
1600     IF Nogo THEN Failexit
1605     Nogo=1
1610     GOTO 1125
1615 MAT X=INV(X)
1620 GOSUB Avgdiags
1625 P=2                                 ![X] IS NOW [Z]2
1630         IF N=2 THEN Finish
1635 GOSUB Partitionx
1640 IF DET(C)<>0 THEN 1660
1645     IF Nogo THEN Failexit
1650     Nogo=1
1655     GOTO 1085
1660 GOSUB Azchange                      ![Z]2 IS CHANGED TO [A]1
1665 P=1                                 ![X] IS NOW [A]1
1670         IF N=1 THEN Finish
1675 GOSUB Partitionx
1680 MAT B=B/(Zo)
1685 MAT C=C*(Zo)
1690 FOR R=1 TO 2
1695 FOR C=1 TO 2
1700     X(R,C)=A(R,C)+B(R,C)+C(R,C)+D(R,C)
1705     X(R,C+2)=A(R,C)-B(R,C)+C(R,C)-D(R,C)
1710     X(R+2,C)=A(R,C)+B(R,C)-C(R,C)-D(R,C)
1715     X(R+2,C+2)=A(R,C)-B(R,C)-C(R,C)+D(R,C)
1720 NEXT C
1725 NEXT R
1730 MAT X=X/(2)
1735 P=5                                 ![X] IS NOW [T]5
1740 GOTO 1395                           !CONTINUE BACKWARD PATH
1745 Azchange: REM [A] TO [Z], AND [Z] TO [A] CHANGES HAPPEN HERE
1750 MAT G=B*C
1755 MAT C=INV(C)
1760 MAT H=A*C
1765 R=C=1
1770 GOSUB Fillx
1775 MAT H=C*D
1780 R=C=3
1785 GOSUB Fillx
1790 MAT H=C
1795 R=3
1800 C=1
1805 GOSUB Fillx
1810 MAT H=A*D
1815 MAT G=H-G
1820 MAT H=C*G
1825 R=1
1830 C=3
1835 GOSUB Fillx
1840 RETURN
1845 Partitionx: REM LOAD 2x2's A,B,C,D FROM X[4,4]
1850 FOR R=1 TO 2
1855 FOR C=1 TO 2
1860 A(R,C)=X(R,C)
1865 B(R,C)=X(R,C+2)
1870 C(R,C)=X(R+2,C)
1875 D(R,C)=X(R+2,C+2)
1880 NEXT C
1885 NEXT R
1890 RETURN
```

```
1895 Fillx: REM LOAD 4 NEW ELEMENTS INTO X[4,4], WITH AVERAGED DIAGS
1900 X(R,C)=X(R+1,C+1)=(H(1,1)+H(2,2))/2
1905 X(R,C+1)=(H(1,2)-H(2,1))/2
1910 X(R+1,C)=-X(R,C+1)
1915 RETURN
1920 Avgdiags: REM AVG DIAGS OF EACH [2,2] SECTION OF [X]
1925 FOR R=1 TO 3 STEP 2
1930 FOR C=1 TO 3 STEP 2
1935 X(R,C)=X(R+1,C+1)=(X(R,C)+X(R+1,C+1))/2
1940 X(R,C+1)=(X(R,C+1)-X(R+1,C))/2
1945 X(R+1,C)=-X(R,C+1)
1950 NEXT C
1955 NEXT R
1960 RETURN
1965 Failexit: REDIM X(5,4)
1970 X(5,1)=P                        !TELLS CALLER OF FAILURE AT THIS POINT
1975 SUBEXIT
1980 Finish: REDIM X(5,4)
1985 X(5,1)=N
1990 Nogo=0                          !SUCCESSFUL EXIT
1995 SUBEND



2000 SUB Source(X(*),Control$,Source$,Gain,R1,R2,Delay)              !############
2005 COM Nogo,Zo,F
2010 IF (Control$="V") AND (Source$="C") THEN Gain=Gain/10^3!TRANS-COND IN mMHOS
2015 IF R1=0 THEN R1=10^(-10)        !KEEPS [Z] WELL BEHAVED
2020 IF R2=0 THEN R2=10^(-10)
2025 MAT X=ZER
2030 X(5,1)=2                        ![Z] FOR THE CONTROLLED SOURCE
2035 X(1,1)=X(2,2)=R1
2040 X(3,3)=X(4,4)=R2
2045 X(3,1)=X(4,2)=-Gain             !ASSUME +I INTO PORT 2 (t=0) WHEN SHORTED
2050 IF (Control$="C") AND (Source$="V") THEN 2065
2055    IF Control$="V" THEN X(3,1)=X(4,2)=R1*X(3,1)
2060    IF Source$="C" THEN X(3,1)=X(4,2)=R2*X(3,1)
2065 IF Delay=0 THEN SUBEXIT         ![Z] IS PURE REAL
2070 Mag=X(3,1)
2075 X(3,1)=X(4,2)=Mag*COS(2*PI*F*Delay/10^3)       !Gain*COS(-wt); t in pico-SEC
2080 X(3,2)=-Mag*SIN(2*PI*F*Delay/10^3)             !jGain*SIN(-wt); t CAN BE NEG
2085 X(4,1)=-X(3,2)                  !Gain FACTOR NOW IN COMPLEX FORM
2090 SUBEND


2095 SUB Rlc(X(*),Type$,R,L,C,Place$)                                 !############
2100 OPTION BASE 1
2105 COM Nogo,Zo,F
2110 DIM A(2,2)                      !TO HOLD THE IMMITANCE OF THE R-L-C
2115 MAT X=ZER
2120 X(1,1)=X(2,2)=X(3,3)=X(4,4)=X(5,1)=1
2125 IF Type$="P" THEN 2175
2130    A(1,1)=A(2,2)=R              !GET Z OF SERIES RLC
2135    A(1,2)=2*PI*F*L
2140    IF C=0 THEN 2155             !MEANS NO CAPACITOR IN THE "RLC" ELEMENT
2145       IF F=0 THEN F=10^(-20)    !APROX D-C
2150       A(1,2)=A(1,2)-10^3/(2*PI*F*C)
2155    A(2,1)=-A(1,2)
2160    IF (Place$="P") AND (DET(A)=0) THEN A(1,1)=A(2,2)=10^(-10)   !SMALL LOSS
2165    IF Place$="P" THEN MAT A=INV(A)
2170    GOTO 2220
2175 REM Type$ IS "P" SO CALCULATE Y
2180    IF R<>0 THEN A(1,1)=A(2,2)=1/R
2185    A(1,2)=2*PI*F*C/10^3
2190    IF L=0 THEN 2205             !NO INDUCTOR
2195       IF F=0 THEN F=10^(-20)
```

```
2200        A(1,2)=A(1,2)-1/(2*PI*F*L)
2205     A(2,1)=-A(1,2)
2210     IF (Place$="S") AND (DET(A)=0) THEN A(1,1)=A(2,2)=10^(-10)
2215     IF Place$="S" THEN MAT A=INV(A)
2220 I=1+(Place$="P")*2
2225 J=3-(Place$="P")*2
2230 X(I,J)=X(I+1,J+1)=A(1,1)       !PUTS THE COMPLEX IMMITANCE INTO [X],
2235 X(I,J+1)=A(1,2)
2240 X(I+1,J)=-A(1,2)               !WHICH IS NOW [ABCD]
2245 SUBEND



2250 SUB Trline(X(*),Zg,Length,K)                                    !############
2255 OPTION BASE 1
2260 COM Nogo,Zo,F
2265 DIM G(4,4),H(4,4)
2270 MAT X=ZER
2275 X(5,1)=4                       ![S] PARAMETERS
2280 Ph=2*PI*F*Length*SQR(K)/11.802854  !PHASE LENGTH OF LINE
2285 X(1,3)=X(2,4)=X(3,1)=X(4,2)=COS(Ph)
2290 X(1,4)=X(3,2)=-SIN(Ph)
2295 X(2,3)=X(4,1)=-X(1,4)
2300 IF Zg=Zo THEN SUBEXIT          ![S] HAS PROPER REFERENCE-Z
2305 REDIM X(4,4)
2310 MAT G=IDN
2315 MAT H=G-X
2320 MAT G=G+X
2325 IF DET(G)=0 THEN G(1,3)=G(2,4)=G(3,1)=G(4,2)=1-10^(-10)
                                    !SMALL LOSS TO AVOID INFINITE [Y]
2330 MAT G=INV(G)                   !CHANGE [S] TO [Y] USING Zg
2335 MAT X=H*G
2340 MAT X=X/(Zg)
2345 REDIM X(5,4)
2350 X(5,1)=3                       ![X] IS NOW [Y]
2355 SUBEND



2360 SUB Tf(X(*),Turns1,Turns2)                                      !############
2365 REM IF TURNS RATIO IS NEG, I AND V ARE FLIPPED AT PORT 2
2370 MAT X=ZER
2375 X(1,1)=X(2,2)=Turns1/Turns2
2380 X(3,3)=X(4,4)=Turns2/Turns1
2385 X(5,1)=1                       ![ABCD] FOR THE TRANSFORMER
2390 SUBEND



2395 SUB Branch(X(*),Type$)                                          !############
2400 OPTION BASE 1                  ![X](5,4) HOLDS BRANCH ELEMENT
2405 COM Nogo
2410 DIM A(2,2),Br(5,4)
2415 MAT Br=X                       !COPY [X] BEFORE RE-DEFINING
2420 MAT X=ZER
2425 X(1,1)=X(2,2)=X(3,3)=X(4,4)=X(5,1)=1
2430 CALL Mtrans(Br(*),2)           ![Br] IS CHANGED TO [Z]
2435 IF Nogo OR (Br(1,1)*Br(1,2)=0) AND (Type$="P") THEN 2510
2440 REM BRANCHES TO DO A SERIES OR PARALLEL OPEN CKT, OR SHORTED SHUNT     NOTE:
A SHORTED SERIES BRANCH IS OK AND NEEDS NO ACTION
2445 IF Type$="P" THEN 2470
2450     X(1,3)=X(2,4)=Br(1,1)      !PUT Z11 FOR THE SERIES BRANCH INTO [X]
2455     X(1,4)=Br(1,2)
2460     X(2,3)=-X(1,4)             ![X] IS NOW [ABCD] FOR THE SERIES BRANCH
2465     SUBEXIT
2470 A(1,1)=A(2,2)=Br(1,1)          !INVERT Z11 TO PUT INTO [X]
```

```
2475 A(1,2)=Br(1,2)
2480 A(2,1)=-A(1,2)
2485 MAT A=INV(A)
2490 X(3,1)=X(4,2)=A(1,1)
2495 X(3,2)=A(1,2)
2500 X(4,1)=-X(3,2)                     ![X] IS NOW [ABCD] FOR THE PARALLEL BRANCH
2505 SUBEXIT
2510    IF Type$="S" THEN X(1,3)=X(2,4)=10^10  !TAKES CARE OF INFINITE-Z SER BR
2515    REM INFINITE-Z PAR BRANCH NEEDS NO CORRECTION TO [X] AS IT IS
2520    IF Nogo=0 THEN X(3,1)=X(4,2)=10^10      !TAKES CARE OF SHORTED PAR BRANCH
2525 SUBEND



2530 SUB Cas(X(*),A(*))                                                      !############
2535 OPTION BASE 1
2540 COM Nogo,Zo,F
2545 DIM Xx(5,4),Aa(5,4)                !TO COPY [X] AND [A] IN CASE BOTH ARE ONE
2550 Pset=1+(X(5,1)+A(5,1)>6)*4         !CHOOSE CLOSER OF ABCD OR [T] PARAMETERS
2555 IF X(5,1)<>Pset THEN CALL Mtrans(X(*),Pset)
2560 IF Nogo=0 THEN 2595
2565    PRINT "CIRCUIT BROKE AT";F;"GHz; SMALL FRONT-TALK IS ASSUMED"
2570    CALL Mtrans(X(*),4)             !GO TO THE LIKELY [S]
2575    IF Nogo THEN PRINT "COULD NOT INTRODUCE FRONT-TALK; ANALYSIS FAULTY"
2580    IF Nogo THEN SUBEXIT            !CAN'T CASCADE ANY MORE TO [X]
2585    X(3,1)=X(4,2)=10^(-10)          !SMALL FRONT-TALK
2590    GOTO 2555
2595 IF A(5,1)<>Pset THEN CALL Mtrans(A(*),Pset)
2600 IF Nogo=0 THEN 2635
2605    PRINT "SOME FRONT-TALK IS ASSUMED TO CASCADE AN ELEMENT AT";F;"GHz"
2610    CALL Mtrans(A(*),4)
2615    IF Nogo THEN PRINT "COULD NOT INTRODUCE FRONT-TALK; ANALYSIS FAULTY"
2620    IF Nogo THEN SUBEXIT            !ELEMENT [A] NOT CASCADED
2625    X(3,1)=X(4,2)=10^(-10)
2630    GOTO 2595
2635 MAT Xx=X
2640 MAT Aa=A
2645 REDIM Xx(4,4),Aa(4,4)
2650 MAT X=Xx*Aa                        !CASCADE [A] ONTO THE RIGHT OF [X]
2655 REDIM X(5,4)
2660 X(5,1)=Pset                        !FINAL ELEMENT IS [ABCD] OR [T]
2665 FOR R=1 TO 3 STEP 2                !AVG DIAGS FOR ACCURACY
2670 FOR C=1 TO 3 STEP 2
2675    X(R,C)=X(R+1,C+1)=(X(R,C)+X(R+1,C+1))/2
2680    X(R,C+1)=(X(R,C+1)-X(R+1,C))/2
2685    X(R+1,C)=-X(R,C+1)
2690    NEXT C
2695    NEXT R
2700 SUBEND



2705 SUB Par(X(*),A(*))                                                      !############
2710 COM Nogo
2715 CALL Mtrans(X(*),3)                !TRY TO GET [Y] FOR THE NETWORK
2720 IF Nogo THEN SUBEXIT               !CAN'T PARALLEL TO AN INFINITE-Y NETWORK
2725 CALL Mtrans(A(*),3)
2730 IF Nogo THEN Failexit
2735 MAT X=X+A                          ![A] IS ADDED TO [X] AND IS UNCHANGED
2740 X(5,1)=3
2745 SUBEXIT
2750 Failexit: MAT X=A                  !NETWORK IS NOW THE INFINITE-Y ELEMENT [A]
2755 SUBEND
```

```
2760 SUB Ser(X(*),A(*))                                          !############
2765 COM Nogo
2770 CALL Mtrans(X(*),2)                !TRY TO GET [Z] FOR THE NETWORK
2775 IF Nogo THEN SUBEXIT               !CAN'T SERIES TO AN INFINITE-Z NETWORK
2780 CALL Mtrans(A(*),2)
2785 IF Nogo THEN Failexit
2790 MAT X=X+A                          ![A] IS ADDED TO [X] AND IS UNCHANGED
2795 X(5,1)=2
2800 SUBEXIT
2805 Failexit: MAT X=A                  !NETWORK IS NOW THE INFINITE-Z ELEMENT [A]
2810 SUBEND


2815 SUB Smith(Xmin,Xmax,Ymin,Ymax)                              !############
2820 GCLEAR
2825 GRAPHICS
2830 FRAME
2835 SHOW Xmin,Xmax,Ymin,Ymax
2840 FOR I=1 TO 7
2845    READ R,R$                       !NORMALIZED RESISTANCE TO BE PLOTTED
2850    Radi=1/(R+1)                    !RADIUS OF CIRCLE
2855    Cen=R*Radi                      !X-COORDINANT OF CENTER
2860    IF R>2 THEN 2875
2865       MOVE Cen-Radi+.015,.015
2870       LABEL R$
2875    MOVE 1,0
2880    FOR L=0 TO 2*PI STEP PI/INT(80/(R+4))
2885       PLOT Cen+Radi*COS(L),Radi*SIN(L)
2890       NEXT L
2895    NEXT I
2900 DATA 0,0,.2,.2,.5,.5,1,1,2,2,5,5,10,10 !VALUES AND NAMES OF R
2905 FOR I=1 TO 6                       !NORMALIZED REACTANCES TO PLOT
2910    READ Th,X$                      !Theta IS <(GAMMA) AT R=0, X$ IS REACTANCE
2915    FOR J=1 TO -1 STEP -2           !PLOT POS & NEG REACTANCE
2920       MOVE COS(Th)+.02-(Th>PI/2)*.13,J*SIN(Th)+J*.04  !LABELING
2925       IF J<0 THEN LABEL "-";
2930       LABEL X$
2935       Radi=TAN(Th/2)
2940       MOVE COS(Th),J*SIN(Th)
2945       S=(2-J)*PI/2                 !USED IN DEFINING ARC LIMITS TO PLOT
2950       FOR L=J*Th+S TO 2*PI-S STEP J*(PI-Th)/INT(16-1/Th)
2955          PLOT 1+Radi*COS(L),Radi*(J-SIN(L))
2960          NEXT L
2965       NEXT J
2970    NEXT I
2975 PLOT -1,0                          !HORIZONTAL LINE
2980 DATA 2.747,.2,2.214,.5,1.571,1,.927,2,.395,5,.199,10  !Th,"x" VALUES
2985 SUBEND


2990 SUB Prt(X(*),Pset)                                          !############
2995 OPTION BASE 1
3000 COM Nogo,Zo,F,Count,SHORT Dat(*)
3005 DIM Mph(2,4),A(2,2),B(2,2),C(2,2) ![Mph] TO HOLD [X] IN MAG,PHASE FORM
3010 Count=Count+1                      !TO STORE NEXT FREQ IN DATA BASE
3015 IF Count>1 THEN 3065               !NO HEADING
3020    IF Pset=1 THEN P$="ABCD"
3025    IF Pset=2 THEN P$=" [Z]"
3030    IF Pset=3 THEN P$=" [Y]"
3035    IF Pset=4 THEN P$=" [S]"
3040    IF Pset=5 THEN P$=" [T]"
3045    PRINT USING 3055;P$
3050    PRINT USING 3060
3055    IMAGE //18X5A" PARAMETERS IN MAGNITUDE AND PHASE"//16X"11"15X"12"15X"2
1"14X"22"9X"K"
```

```
3060    IMAGE XX"FREQ"6X"MAG"4X"ANG"2(7X"MAG"4X"ANG")6X"MAG"4X"ANG
"4X"FACT"
3065 IF X(5,1)<>Pset THEN CALL Mtrans(X(*),Pset)
3070 IF Nogo THEN Failexit
3075 Dat(Count,1)=F                   !LOAD 1 FREQ INTO DATA BASE
3080 FOR C=2 TO 9
3085    Dat(Count,C)=X(1+(C>5)*2,C-1-(C>5)*4)
3090    NEXT C
3095 Dat(101,1)=Pset                  !TYPE OF PARAMS BEING LOADED IN DATA BASE
3100 FOR R=1 TO 3 STEP 2
3105 FOR C=1 TO 3 STEP 2
3110    Mph(R-(R=3),C)=SQR(X(R,C)^2+X(R,C+1)^2)
3115    IF Pset=3 THEN Mph(R-(R=3),C)=Mph(R-(R=3),C)*10^3 ![Y] PRINTED IN mMHOS
3120    IF X(R,C)*X(R,C+1)=0 THEN Mph(R-(R=3),C+1)=(SGN(X(R,C))*(SGN(X(R,C))-1)+
SGN(X(R,C+1)))*90
3125    IF X(R,C)*X(R,C+1)<>0 THEN Mph(R-(R=3),C+1)=180*ATN(X(R,C+1)/X(R,C))/PI-
(SGN(X(R,C))-1)*SGN(X(R,C+1))*90
3130    NEXT C
3135    NEXT R
3140 CALL Mtrans(X(*),3)              !GET [Y] FOR CALC OF K FACTOR
3145 FOR I=1 TO 2
3150 FOR J=1 TO 2
3155    A(I,J)=X(I,J+2)
3160    B(I,J)=X(I+2,J)
3165    NEXT J
3170    NEXT I
3175 MAT C=A*B
3180 K=(2*X(1,1)*X(3,3)-C(1,1))/SQR(DET(C))
3185 IF (Pset=2) OR (Pset=3) THEN 3210
3190 IF Pset=1 THEN 3225
3195    PRINT USING 3200;F,Mph(*),K
3200    IMAGE 3D.3D,4D.4D,2(5D.D,5D.4D),5D.D,4D.4D,5D.D,4D.2D
3205    SUBEXIT
3210 PRINT USING 3215;F,Mph(*),K
3215 IMAGE 3D.3D,6D.2D,5D.D,2(7D.2D,5D.D),6D.2D,5D.D,4D.2D
3220 SUBEXIT
3225    Mph(2,1)=Mph(2,1)*10^3        !CONDUCTANCE PRINTED IN mMHOS
3230    PRINT USING 3235;F,Mph(*),K
3235    IMAGE 3D.3D,4D.4D,5D.D,2(6D.3D,5D.D),4D.4D,5D.D,4D.2D
3240    SUBEXIT
3245 Failexit: PRINT USING 3250;F,P$
3250    IMAGE 3D.3D,3X5A"_PARAMETERS CANNOT BE CALCULATED AT THIS FREQUENCY"
3255    FOR I=2 TO 9
3260       Dat(Count,I)=0             !DATA LOST AT THIS FREQ
3265       NEXT I
3270 SUBEND



3275 SUB Splot(I,J)                                               !############
3280 COM Nogo,Zo,F,Count,SHORT Dat(*)
3285 IF Dat(101,1)<>4 THEN CALL Dtrans(4)
3290 C=2+(I>1)*4+(J>1)*2              !COL OF DATA BASE WITH REAL PART OF S(I,J)
3295 MOVE Dat(1,C),Dat(1,C+1)
3300 FOR R=1 TO 100                   !ENTIRE DATA BASE
3305    IF (R>1) AND (Dat(R,1)=0) THEN SUBEXIT  !END OF STORED PARAMS
3310    PLOT Dat(R,C),Dat(R,C+1)      !DATA BASE PLOTED AS IS: REAL,IMAG FORM
3315    NEXT R
3320 SUBEND
```

```
3325 SUB Flip(X(*))                                                    !############
3330 OPTION BASE 1
3335 COM Nogo,Zo,F
3340 DIM Xx(4,4),G(4,4)
3345 Pset=X(5,1)
3350 REDIM X(4,4)
3355 IF (Pset>1) AND (Pset<5) THEN 3435 !DO DIAGONAL FLIPPING
3360    IF DET(X)<>0 THEN 3390
3365       REDIM X(5,4)
3370       CALL Mtrans(X(*),4)       !TRY FOR [S]
3375       IF Nogo=0 THEN 3430
3380          PRINT "PORTS CANNOT BE INTERCHANGED IN";Pset;"OR [S] AT";F;"GHz"
3385          SUBEXIT
3390    MAT X=INV(X)
3395    IF Pset=5 THEN 3435          !DO DIAG FLIPPING
3400    G(1,1)=G(2,2)=1
3405    G(3,3)=G(4,4)=-1
3410    MAT Xx=G*X
3415    MAT X=Xx*G
3420    REDIM X(5,4)                 !X(5,1) RETAINS ORIG VALUE
3425    SUBEXIT
3430 REDIM X(4,4)
3435 G(1,3)=G(2,4)=G(3,1)=G(4,2)=1 !DIAG FLIPPING FOR [Z] [Y] & [S]
3440 MAT Xx=G*X
3445 MAT X=Xx*G
3450 REDIM X(5,4)
3455 SUBEND


3460 SUB Dtrans(Pset)                                                  !############
     CONVERTS ENTIRE DATA BASE TO Pset WHILE PRINTING NEW PARAMS
3465 OPTION BASE 1
3470 COM Nogo,Zo,F,Count,SHORT Dat(*)
3475 DIM X(5,4)                      !TEMP HOLDER
3480 Count=0                         !TO SIMULATE AN ANALYSIS RUN OF THE CKT
3485 Oldpset=Dat(101,1)              !CURRENT PARAM TYPE
3490 FOR I=1 TO 100
3495    IF (I>1) AND (Dat(I,1)=0) THEN SUBEXIT  !END OF STORED DATA
3500    FOR R=1 TO 3 STEP 2          !LOAD [X] WITH 1 FREQ OF [Dat]
3505    FOR C=1 TO 3 STEP 2
3510       X(R,C)=X(R+1,C+1)=Dat(I,2+(R>1)*4+(C>1)*2)
3515       X(R,C+1)=Dat(I,3+(R>1)*4+(C>1)*2)
3520       X(R+1,C)=-X(R,C+1)
3525       NEXT C
3530       NEXT R
3535    X(5,1)=Oldpset
3540    F=Dat(I,1)
3545    CALL Prt(X(*),Pset)          !PRINTS AND STORES NEW PARAMS (IF EXIST)
3550    NEXT I
3555 SUBEND



3560 SUB Lossuline(X(*),Zgo,Length,K,Ac,Ad,Fo)                        !############
     UNITS ARE OHMS,INCHES,RELATIVE EPSILON,dB/IN,dB/IN,GHz
3565 OPTION BASE 1
3570 COM Nogo,Zo,F
3575 DIM Zg(2,2),Gamma(2,2),Zpl(2,2),Ypl(2,2),G(4,4),H(4,4)
3580 MAT X=ZER
3585 Ac=Ac*.11513                    !CONVERTS TO NEPER/IN BY ln10 NEPERS=20dB
3590 Ad=Ad*.11513
3595 Wo=2*PI*Fo
3600 L=Zgo*SQR(K)/11.802854          !INDUCTANCE PER INCH (nH)
3605 C=L/Zgo^2                       !CAPACITANCE/IN (nF)
3610 R=SQR(F/Fo)*2*Ac*(Ac+SQR(2*Ac^2+Wo^2*L*C))/(Wo*C)   !COPPER RESIST/IN
3615 G=F*2*Ad*SQR(Ad^2+Wo^2*L*C)/(Wo*L*Fo)  !DIELECTRIC CONDUC/IN
```

```
3620     Zpl(1,1)=Zpl(2,2)=R
3625     Zpl(1,2)=2*PI*F*L+R
3630     Zpl(2,1)=-Zpl(1,2)
3635     Ypl(1,1)=Ypl(2,2)=G
3640     Ypl(1,2)=2*PI*F*C
3645     Ypl(2,1)=-Ypl(1,2)
3650 MAT Gamma=Zpl*Ypl             !IS GAMMA SQUARED
3655     Mag=SQR(SQR(DET(Gamma)))
3660     Ph=ATN(Gamma(1,2)/Gamma(1,1))/2
3665     IF Ph<=0 THEN Ph=Ph+PI/2    !TO GET 0<ANGLE(Gamma)<=PI/2
3670     Alpha=Mag*COS(Ph)
3675     Beta=Mag*SIN(Ph)
3680 MAT Ypl=INV(Ypl)
3685 MAT Zg=Zpl*Ypl                !IS GUIDE-IMPEDANCE SQUARED
3690     Mag=SQR(SQR(DET(Zg)))
3695     Ph=ATN(Zg(1,2)/Zg(1,1))/2   ! -PI/4<Ph<PI/4
3700     Zg(1,1)=Mag*COS(Ph)
3705     Zg(1,2)=Mag*SIN(Ph)
3710 Alpha=EXP(-Alpha)
3715 X(1,3)=X(2,4)=X(3,1)=X(4,2)=Alpha*COS(Beta*Length)
3720 X(1,4)=X(3,2)=-Alpha*SIN(Beta*Length)
3725 X(2,3)=X(4,1)=-X(1,4)
3730 X(5,1)=4                      ![X] IS [S] PARAMS IN A COMPLEX Zg SYSTEM
3735 REDIM X(4,4)                  !BEGIN CHANGE TO [Z] PARAMS
3740 MAT G=IDN
3745 MAT H=G-X
3750 IF DET(H)<>0 THEN 3770
3755     PRINT "LINE HAS NEGLIGIBLE LOSS AND HALF-WAVELENGTH; [S] HAS WRONG Zref"
3760     REDIM X(5,4)
3765     SUBEXIT
3770 MAT H=INV(H)
3775 MAT G=G+X
3780 MAT X=G*H
3785 MAT G=X
3790 MAT H=ZER
3795 H(1,1)=H(2,2)=H(3,3)=H(4,4)=Zg(1,1)
3800 H(1,2)=H(3,4)=Zg(1,2)
3805 H(2,1)=H(4,3)=-H(1,2)
3810 MAT X=G*H                     !DENORMALIZES FROM Zg
3815 REDIM X(5,4)
3820 X(5,1)=2                      ![Z] PARAMS FOR THE LOSSY LINE
3825 SUBEND


3830 SUB Pread(X(*))                                                 !############
     LOADS X(5,4) WITH (INTERPOLATED) PARAMS FROM TYPED-IN DATA
3835 OPTION BASE 1
3840 COM Nogo,Zo,F
3845 DIM F(50),Coef(3),M(3,3),V(3),Buf(8),P(3,8) ![Buf] HOLDS PARAMS AT 1 FREQ;
     [P] HOLDS PARAMS AT 3 STRADDLING FREQS, F IN UPPER SEG IF POSS
3850 READ Pset,N
3855 MAT READ F(N)                 !REDIMS [F] TO USER'S N FREQS
3860 IF N>2 THEN 3960
3865 IF N=2 THEN 3890
3870     MAT READ P(1,8)
3875     GOSUB Convert             !CHANGE FROM MAG,PH TO REAL,IMAG
3880     MAT Buf=P                 !PARAMETERS TAKEN AS CONST IF GIVEN 1 FREQ
3885     GOTO Loadpars
3890 MAT READ P(2,8)
3895 GOSUB Convert
3900 REDIM Coef(2),M(2,2),V(2)     !DO A LINEAR INTERP WITH 2 FREQS
3905 M(1,1)=M(2,2)=1
3910 M(1,2)=F(1)
3915 M(2,2)=F(2)
3920 MAT M=INV(M)
3925 FOR C=1 TO 8
```

```
3930    V(1)=P(1,C)                   !VECTOR OF PARAMS (REAL OR IMAG) AT 2 FREQS
3935    V(2)=P(2,C)
3940    MAT Coef=M*V
3945    Buf(C)=Coef(1)+Coef(2)*F
3950    NEXT C
3955 GOTO Loadpars
3960 FOR I=1 TO N                     !BEGIN GEN SEARCH FOR 3 FREQS AROUND F
3965    IF F(I)=F THEN Usedata        !HAVE DATA AT EXACT FREQ; NEED NO INTERP
3970    IF F(I)>F THEN 3980
3975    NEXT I
3980 Skip=MIN(I-3,N-3)                !#FREQS NOT USED FOR THIS INTERP
3985 Skip=MAX(Skip,0)
3990 FOR I=1 TO Skip*8
3995    READ Waste                    !IGNORE UNWANTED FREQ DATA
4000    NEXT I
4005 READ P(*)
4010 GOSUB Convert
4015 M(1,1)=M(2,1)=M(3,1)=1           !BEGIN SOL'N FOR PARABOLIC INTERPOLATION
4020 FOR I=1 TO 3
4025    M(I,2)=F(Skip+I)              !MATRIX OF 1,F,F^2 FOR 3 FREQS
4030    M(I,3)=M(I,2)^2
4035    NEXT I
4040 MAT M=INV(M)                     !WILL EXIST IF FREQS ARE DIFF
4045 FOR C=1 TO 8                     !4 MAGS AND PHASES TO INTERPOLATE
4050    V(1)=P(1,C)
4055    V(2)=P(2,C)
4060    V(3)=P(3,C)
4065    MAT Coef=M*V                  !a,b,c IN a+bF+cF^2
4070    Buf(C)=Coef(1)+Coef(2)*F+Coef(3)*F^2   !THE PREDICTED VALUE AT DESIRED F
4075    NEXT C
4080 GOTO Loadpars
4085 Usedata:FOR J=1 TO (I-1)*8
4090    READ Waste
4095    NEXT J
4100 READ Buf(*)
4105 Loadpars:X(5,1)=Pset             !BEGIN TO LOAD [X]
4110 FOR I=1 TO 3 STEP 2              !4 PARAMETERS TO LOAD INTO [X]
4115 FOR J=1 TO 3 STEP 2
4120    C=J+(I=3)*4                   !COL OF BUFFER WITH X(I,J) (REAL PART)
4125    X(I,J)=X(I+1,J+1)=Buf(C)
4130    X(I,J+1)=Buf(C+1)
4135    X(I+1,J)=-X(I,J+1)
4140    NEXT J
4145    NEXT I
4150 SUBEXIT
4155 Convert: FOR R=1 TO ROW(P) !CHANGES ALL ROWS OF [P] FROM MAG,PH TO REAL,IM
4160 FOR C=1 TO 7 STEP 2
4165    Mag=P(R,C)
4170    Ph=P(R,C+1)*PI/180
4175    P(R,C)=Mag*COS(Ph)
4180    P(R,C+1)=Mag*SIN(Ph)
4185    NEXT C
4190    NEXT R
4195 RETURN
4200 ! PUT HERE:DATA Pset,#Freqs,F1...FN (INCREASING FREQS ONLY!)
4205 ! DATA IN Mag,Phase(IN DEGREES) FOR EACH PARAMETER 11,12,21,22 AT EACH FREQ
4210 !
4215 !
4220 !
4225 !
4230 !
4235 !
4240 !
4245 !
4250 !
4255 SUBEND
```