27
11-1-82

I-6200

①

Dr. 954

UCID- 19563

ELECTRONICS ENGINEERING DEPARTMENT

Computer Program Newsletter
Number 7

W. G. Magnuson, Jr.

September 1982

Lawrence
Livermore
Laboratory

Work performed under the auspices of the U.S. Department of Energy by the Lawrence Livermore Laboratory under Contract W-7405-Eng-48.

UCID--19563

DE83 001845

ELECTRONICS ENGINEERING DEPARTMENT

Computer Program Newsletter
Number 7

Electronic Circuit Analysis Using Computer Programs*

September 1982

W. G. Magnuson, Jr.

Abstract

This issue of the Computer Program Newsletter updates an earlier newsletter (Number 2, September 1979) and focuses on electrical network analysis computer programs. In particular, five network analysis programs (SCEPTRE, SPICE2, NET2, CALAHAN, and EMTP) will be described. The objective of this newsletter will be to provide a very brief description of the input syntax and semantics for each program, highlight their strong and weak points, illustrate how the programs are run at Lawrence Livermore National Laboratory using the Octopus computer network, and present examples of input for each of the programs to illustrate some of the features of each program. In a sense, this newsletter can be used as a quick reference guide to the programs.

Contents Page

1. Introduction - Historical Development of Circuit Analysis Programs

Although the subject area of circuit analysis is an old one in the subject domain of electrical engineering, the development of digital computer programs for solving systems of equations describing electrical networks is relatively recent. The first digital computer programs for electrical circuit analysis were formulated in the late 1950s. Very specific programs were written then, to solve for particular network topologies. A few programs were developed to describe electrical filter response and to do some synthesis.

The foundations for analyzing a more general network had been laid much earlier by Gabriel Kron in his 1939 book, Tensor Analysis of Networks. Kron's book presented Kirchhoff's laws and network topology as matrices which was an almost ideal representation for digital computer implementation. Frank Branin in the late 1950's and early 1960's employed Kron's methods to develop the program TAP (Transistor Analysis Program) which was one of the first generation programs (1962). Branin was primarily the one responsible for implementing the matrix algebraic topological techniques of Kron[ ]. While Branin (at IBM) was developing TAP, Ashcraft and Hochwalt at Autonetics were developing SPARC and SCAN which resulted in the general program TRAC in 1963.

I think because of the development of TAP at IBM, the Air Force Weapons Laboratory at Albuquerque, New Mexico issued a contract with IBM in the summer of 1962 to develop a general-purpose circuit analysis program. This program, called PREDICT, was to have features in it for the determination of circuit responses in radiation environments. PREDICT was released in the summer of 1964.

TAP also initiated the development of the ECAP program at IBM. ECAP became available in the fall of 1964 and became a standard in industry and universities for comparing circuit analysis programs. Partly because Branin had worked at the Los Alamos Scientific Laboratory in the mid-1950's, a program development was started there by Allan Malmberg which produced the NET-1 program in 1964. During the development of NET-1 a summer visit by Richard Dickhaut of Boeing Aircraft to Los Alamos initiated the start of the CIRCUS program at Boeing.

These programs (PREDICT, NET-1, and CIRCUS) all were aimed at analyzing electronic circuits subjected to a radiation environment. In the mid-1960's several other programs also appeared on the scene (CIRCAL, CORNAP, CALAHAN, LISA) and represented an era of different analysis methods and different numerical techniques being used. Nodal, state variable, topological, and hybrid network formulation techniques were employed.

In the late 1960's and early 1970's several improvements in numerical and matrix methods were incorporated into analysis programs. Implicit algorithms for solving networks characterized by stiff state equations appeared. These algorithms essentially solved one of the problems which had plagued early analysis programs-networks with a wide spread in circuit time constants. Sparse matrix methods were developed by electrical power industry at Bonneville Power Administration in the mid 1960's to handle large electrical distribution networks. These techniques were incorporated in most other circuit analysis programs in the late 1960's to deal with larger networks and to speed solutions.

These improvements led to the second generation of programs such as ECAP-II, SCEPTRE, SCEPTRE-II, ASTAP, NET-2, CIRCUS2, SPICE, and others. Most of the severe technical limitations of the earlier programs have been overcome.

In the past few years, the emphasis in circuit analysis computer program development has been aimed at increased modeling sophistication to more properly account for device physics and at greatly increasing circuit size capability so as to deal with increasingly complex circuits (VLSI). Optimization, worst case, Monte Carlo, sensitivitiy, mixed-mode (analog-digital), and other analysis features have increasingly become part of analysis program capbilities.

2. Historical Perspective of Electronic Circuit Analysis by Computers at LLNL.

The first circuit analysis program available at LLL was an early version of the ECAP program which became available on an IBM 7094 computer in October 1964. In December of 1964, the PREDICT program (an early version of the SCEPTRE program) became available also on the IBM 7094 computer. In early 1965 through 1969, courses were taught on the use of these and other circuit analysis programs. The SCEPTRE program replaced PREDICT in 1966.

In the period 1965 to 1970, several additional programs were implemented on the CDC 6600 and 7600 computers. These programs included both interactive and batch processing programs and found limited but rather continuous use. CALAHAN, POTTLE, CORNAP, BIAS, SINC, SLIC, SPICE, CIRCUS, NASAP, GINA were some of the programs available.

For the most part, all of these programs provided either dc, ac, a transient analysis or a combination of these three types of analysis. Each program had its own input format, a situation which still exists to a large extent among circuit analysis programs.

Although some of the programs mentioned earlier are still capable of being run on the CDC 7600 computer systems, the programs described in this newsletter provide a good cross-section of capabilities.

Currently available on the CDC 7600 computers at LLNL are the SCEPTRE, NET-2, SPICE2, CALAHAN, and EMTP programs. SPICE2 and EMTP also run on the Cray-1 computer and SCEPTRE-II is in the process of being rewritten for the Cray-1. SPICE2 is also operational on the VAX computer at LLNL.

3. Program Descriptions

In this section, five electrical circuit analysis programs will be described. All five programs are available on the LLNL Octopus computer network. Each of the programs offer some unique capability or feature which might make it the choice for a particular analysis.

The descriptions for each program will pursue the following outline:

1. Purpose and General Description,

2. Limitations and Applicability,

3. Availability and Operation,

4. Input Format,

5. LLNL Contact,

6. References.

The input format section will be very abbreviated and should not be considered as describing all facets of the programs. Expanded program descriptions would be found in the references. Examples of the input format are deferred to section 4 where illustrative examples are given for each of the programs.

The examples were selected to illustrate typical problems and features of the programs. No output is included because it would add greatly to the length of this Newsletter. To obtain output, simply run the appropriate example and send the output to a printer.

To use any of the circuit analysis programs, several steps must be taken. Although the details may vary depending on which program is used, the general steps are:

1. Draw an equivalent circuit comprising of resistors, capacitors, inductors, voltage and current sources, etc.

2. Assign a name or number to all nodes in the circuit.

3. Give a name or number to each circuit element.
4. Assume current flow directions in each element and source.
5. Choose circuit values in a consistent set of parameters units.
6. Prepare the input using an online editor.
7. Run the analysis program.
8. Look at and analyze the output.

3.1 SCEPTRE

3.1.1 Purpose and General Description

SCEPTRE is a general-purpose electronic engineering compter aided design program available for execution on all CDC 7600 computers under the Livermore time-sharing system. The program is designed to determine the initial conditions and/or transient response of electronic circuits.

Among the many useful features and options available to the user are stored model capability, automatic initial condition determination, automatic reruns with a minimum of input modification, a parameter definition to allow output of quantities other than sources or passive currents and voltages, and direct entry of differential equations.

Particularly useful in the SCEPTRE program is the extremely flexible way in which network elements may be modeled. SCEPTRE allows table descriptions, mathematical equations, or combinations of tables and equations to represent the time behavior of most any element. Fortran SUBPROGRAMS may also be included in the input to add to the flexibility of modeling.

3.1.2 Limitations and Applicability

The program data limits are adequate for most circuit analysis work. Storage is dynamic in SCEPTRE but the following are approximate limits:

Program Data Limits

Description of Data	Maximum Number
Heading Cards	11
Elements	300
Nodes	301
Source Derivatives	50
Defined Parameter Differential Equations	100
Defined Parameter Differential Equations	100
Mutual Inductances	50
Arguments in Equation Value Specification	50
Model Table Changes	15
Model Equation Changes	15
Model Output Suppressions	10
Output Requests	100
Supplied Initial Conditions	100
Equation Functions (1 equation per card)	80 (approx.)
Cards per Equation Function	20
Table Functions	80 (approx.)
Optional Termination Conditions	10

Models on Library Tape (Combined)	250
Characters in Model Name	18
Model Terminals (External Nodes)	25
Model Internal Nodes	301

The name for nodes, elements, equations, models also have limits. Frequently, puzzling results stem from exceeding alphanumeric character lengths which are:

Maximum Alphanumeric Character Lengths Allowed

Item	Circuit Description Maximum Number	Model Description Maximum Number[1]
Nodes Names	6	3
Element Names	5	2
Defined Parameter Names	6	3
Table Names	5	2
Equation Names	5	2
Model Names	18	18
Output Labels	6	3
Circuit Designation (for calling models)	3[1]	-

Note:
1. Recommended

In addition to the above limits, there are a few topological restrictions. Voltage source loops and current source cut sets are not allowed. Likewise for dc convergence, loops with only voltage sources and inductors and cut sets with only current sources and capacitors should be avoided. DC convergence may not be obtained if resistor or inductor current is used as an independent variable for functional element value specification. Capacitor voltages may be used if the capacitor is in parallel with a resistor or current source.

Computer run time is usually not a serious factor with SCEPTRE runs. Input data is read and interpreted by the first phase of the program and if no fatal errors are detected, a FORTRAN program is written. The FORTRAN program is then compiled and the object code is executed. The majority of all analyses requires less than one minute of CDC 7600 computer time.

3.1.3 Availability and Operation

SCEPTRE is the name of the network analysis program. The actual process of making a computer run with SCEPTRE is rather involved so to make it more convenient, a controller program is used. This controller program is called SCEPTRE.

The execute line (on CDC 7600 computers) is:

```
SCEPTRE INFILE OUTFILE MODELFILE BOX NXX IDENTIFICATION / T V
```

SCEPTRE is stored in the CDC 7600 R and U disk file EELIB. So the initially run of SCEPTRE should be:

```
X EELIB SCEPTRE etc.
```

Any or all arguments may be dropped out from the right.

Argument	Description	Default
INFILE	SCEPTRE input source (prepared by user)	None
OUTFILE	Listing output	RESULTS
MODELFILE	Permanent model library (currently unused)	MODLIB
BOX		None
NXX	Your box number	None
IDENTIFICATION	22-char identification line	None

INFILE, BOX NXX IDENTIFICATION should be on the input line. If they are not, they will be requested.

If errors are detected during the translation of your input INFILE, an error message will print on the TTY, and execution of SCEPTRE will terminate. See the output file (default - RESULTS) for the actual error code and then refer to the list of error statements in the user manual for the actual error message. An additional file, BUG, is written by the SCEPTRE controller and should also be inspected if an error occurs during the run. BUG frequently indicates what actually caused the error when the error is not obvious in the output file.

3.1.4 Input Format

The SCEPTRE circuit description language is a user-oriented free-format language. Only information is accepted in columns 1 through 72. Information can be anywhere in columns 1-72 and the use of delimiters makes the input easy to interpret. The SCEPTRE language makes use of major headings and subheadings (it is block structured). The appropriate information must be placed under its proper heading. The headings and subheadings are:

```
MODEL DESCRIPTION (INITIAL, PRINT)

     MODEL NAME (PERM or TEMP) (NODE-NODE-...-NODE)
     (Comment cards, up to 11 allowed)

          ELEMENTS
          DEFINED PARAMETERS
          OUTPUTS
          FUNCTIONS

     CIRCUIT DESCRIPTION
     (Comment cards, up to 11 allowed)

          ELEMENTS
          DEFINED PARAMETERS
          OUTPUTS
          INITIAL CONDITIONS
          FUNCTIONS
          RUN CONTROLS

     RERUN DESCRIPTION (N)

     (Comment cards, up to 11 allowed)
          ELEMENTS
          DEFINED PARAMETERS
          INITIAL CONDITIONS
          FUNCTIONS
          RUN CONTROLS

     SUBPROGRAM

          FUNCTION or SUBROUTINE NAME (arguments)
          (FORTRAN statements)
          END

     END
```

The SUBPROGRAM, MODEL DESCRIPTION, and RERUN DESCRIPTION information will not be used for most runs and need not be included.

The information under each of the subheadings contains the circuit describing information making use of symbols and punctuation. The SCEPTRE manual makes use of the following definitions:

ELEMENT NAME - Denotes the name given to each component (including model circuit designations) of a circuit (e.g., RA, LLX, E17). No more than five alphanumeric characters may be used to name an element. Model circuit designations are limited to no more than four alphanumeric characters.

NODE - Denotes the designation assigned to each node of a circuit. No more than six alphanumeric characters may be used to name a node.

NUMBER - A numerical constant that may be written as a signed quantity in either integer or decimal form and with or without an exponent. Up to 13 characters may be used to represent a number. For example, numbers may be written in the following forms: 10, 10., 10.0, -.1, -0.1, +1.4, 6.4E9, -74.3E-7, 7E+11, -176.6667E5.

CONSTANT - Same as NUMBER except a decimal point must be included in the specification of the numerical constant.

VALUE - Will be used to denote any of the following: NUMBER, DEFINED PARAMETER, TABLE, EQUATION or Mathematical Expression.

SPECIAL VALUE - Will be used to denote any of the following: VALUE, CONSTANT * Resistor Current, CONSTANT * Resistor Voltage, VALUE * Current Source, DIODE TABLE or DIODE EQUATION (X1, X2).

VARIABLE - Denotes any of the following:

1. The voltage or current associated with any element as VR1, VJ7, IE4, ILM, etc.
2. Any source or source derivative as J17, DJ17, E4, DE4, etc.
3. Any defined parameters as P7, DP7, etc.
4. Any element value as R17, CA, M12, etc.
5. Time as TIME.
6. Any internal parameter.

V ELEMENT NAME or I ELEMENT NAME - Denotes the element voltage or current of ELEMENT NAME. For example, the voltage across capacitor CAB1 would be referred to as VCAB1, and the current through inductor LCHOK would be referred to by ILCHOK.

TABLE NAME (INDEPENDENT VARIABLE) - Used when a variable circuit quantity is given in tabular form. The table used must be given a unique name prefixed by TABLE or simply T, and followed by a single independent variable in parenthesis. The name may consist of up to five alphanumeric characters. The independent variable may be any of the quantities defined under VARIABLE, such as, TABLE 1A (VC1). If an independent variable, including the enclosing parenthesis is not supplied, then TIME will automatically be chosen.

EQUATION NAME (ARGUMENT LIST) - Used when a variable circuit quantity is given in closed form. The equation must be given a name prefixed by EQUATION or simply Q and followed by one or more arguments separated by commas and enclosed in parentheses. The EQUATION name may consist of up to five alphanumeric characters. The argument list may consist of any VARIABLE, CONSTANT and TABLE (and its independent variable). For example, EQUATION 39 (VCX, J2, TIME, TABLE 2 (VC7)).

The full description of all of the features and options of the SCEPTRE is beyond the intent of this newsletter. Instead of trying to describe all features, only three basic subheading information groups will be covered. A minimum input set for a SCEPTRE run would consist of the following:

```
CIRCUIT DESCRIPTION
          (Comments)
     ELEMENTS
          (circuit description)
     OUTPUTS
          (variables to be plotted)
     RUN CONTROLS
          (analysis options)
END
```

The information under the ELEMENTS, OUTPUTS, and RUN CONTROLS will be very briefly described in the following.

ELEMENTS

All circuit elements (resistors, capacitors, inductances including mutual, voltage and current sources, source derivatives, and model designations) are input under this subheading. This node-to-node input defines the topology of the network to be analyzed. The general form is:

ELEMENT NAME, NODE - NODE = SPECIAL VALUE

The first character of the ELEMENT NAME designates the type of network element. Entries that are allowed are summarized below.

Entries Under Elements

Name		Nodes		Value Specification
R name	,	NODE-NODE		NUMBER
C name	,	NODE-NODE		TABLE name (Independent Variable)
L name	,	NODE-NODE		
E name	,	NODE-NODE	= VALUE	Defined Parameter
J name	,	NODE-NODE		EQUATION name (Argument List)
M name	,	L NAME-L NAME		EXPRESSION name (Mathematical definition)
				EXTERNAL FUNCTION (Argument List)

Linearly Dependent Sources

E name	,	NODE-NODE	= CONSTANT * VR name
J name	,	NODE-NODE	= CONSTANT * IR name

Primary Dependent Current Sources

J name	,	NODE-NODE	= DIODE TABLE name DIODE EQUATION (X1, X2)

Secondary Dependent Current Sources

J name	,	NODE-NODE	= (VALUE * J name) (J is a primary dependent current source)

Voltage and Current Source Derivatives

DE = VALUE

DJ = VALUE

Model Calls

Circuit Designation		Nodes	Model Name
D name	,	NODE-NODE	= MODEL 1NXXX (PERM)
T name	,	NODE-NODE-NODE	= MODEL 2NXXX (TEMP)
AX1	,	NODE-NODE-...NODE	= MODEL XYZ (TEMP)

OUTPUTS

The general format for specifying SCEPTRE output is:

VARIABLE, VARIABLE, VARIABLE, PLOT

VARIABLE has been defined earlier. Considerable flexibility is allowed in relabeling or plotting one variable versus another. The SCEPTRE Manual should be referred to for these options.

RUN CONTROLS

The information in the section controls the run but does not directly affect the network. A minimum entry under this section would be

STOP TIME = NUMBER

which would specify the amount of analysis time desired for a transient analysis to run. If tabular printed output is desired in addition to plots then use

MAXIMUM PRINT POINTS = NUMBER

3.1.5 LLL Contact

The best source for information is the SCEPTRE User's Manual. The peculiarities of running the SCEPTRE program on the OCTOPUS system can be answered by Waldo Magnuson, Ext. 2-9950, Bldg. 131, Rm. 2638. Waldo can also help on problems which arise in circuit input preparation or circuit modeling. Copies of the items listed in the references are available by contacting Waldo.

3.1.6 References

1. S. R. Sedore and J. R. Seuts, "SCEPTRE SUPPORT II, Volume I: Revised SCEPTRE User's Manual," AFWL-TR-69-77, Vol. I, July 1970.
2. SCEPDOC (Available from Library file EELIB by typing X EELIB SCEPDOC DR.)

3.2 SPICE2

3.2.1 Purpose an General Description

SPICE2 is a circuit simulation program for nonlinear DC, nonlinear transient, and linear AC analyses. Circuits may contain linear resistors, capacitors, inductors, mutual inductors, independent voltage and current sources, four types of dependent sources, transmission lines, and four nonlinear semiconductor devices: diodes, BJTS, JFETS, and MOSFETS.

SPICE2 has built-in models for the semiconductor devices and the user need specify only the pertinent model parameters. The program, developed at the University of California, Berkeley, is intended primarily for aiding in the design of integrated circuit chips. However, because of the program's generality, it is frequently useful in analyzing a broad range of linear and electronic circuits.

There are several potentially useful analysis options for SPICE2. They include:

DC analysis,

AC small-signal analysis,

Transient analysis,

Analyses at different temperatures,

DC small-signal sensitivity analysis,

AC small-signal distortion analysis,

Noise analysis, and

Fourier analysis as part of the transient analysis.

Besides the built-in models for semiconductor devices, SPICE2 has a subcircuit modeling capability. This is frequently useful for networks which are repetitive in nature.

3.2.2 Limitations and Applicability

Central memory in SPICE2 is allocated dynamically to correspond to the needs of the simulation. The size at LLL can expand to 156000 octal words, which is close to the maximum small core memory size allowed. All lists, tables, and arrays are referenced by pointers. The largest program overlay is DCTRAN and requires about 6000 octal words. Thus circuit describing data has considerable room to expand as needed and the only limitation imposed by the program on the size or the complexity of the circuit to be simulated is that all necessary data fit in memory.

Both DC and transient solutions are obtained by an iterative process which terminates whenever the nonlinear branch currents converge to whitin a tolerance of 0.1 percent and the node voltages converge to within 0.1 percent.

Although the SPICE2 algorithms are reliable, in some cases convergence does not occur. When this happens, the program stops with a message and, in such cases, the node voltages are not necessarily even close to the correct solution. Regenerative switching circuits, circuits with positive feedback, and networks with very large values for sources are likely candidates for failure to converge.

3.2.3 Availability and Operation

The SPICE2 program is available for execution on the Cray-1 C and D computers and the R and U CDC 7600 computers. The executable code is named SPICE2 and may be obtained from the library file EELIB by typing:

```
X EELIB SPICE2 DR./ t v
```

The user's guide is named SPICEGUIDE and a copy is available by typing:

```
XPORT RD .558850:SPICE:2G.5:SPICEGUIDE
```

<u>Cray-1 SPICE2 Execution</u>

The SPICE2 program may be directly on the Cray-1 computers. The execution line is:

```
SPICE infilename outfilename / t v
```

where infilename and outfilename are input and output file names. Default names are INPUT and OUTPUT. After execution the output may be viewed on-line with TRIX AC or TRIX GL or sent to an RJET printer using ALLOUT (use LONG. and COL1. options).

<u>CDC 7600 SPICE2 Execution</u>

SPICE2 runs under the SLOPE system on the CDC 7600 computers. The input data file <u>must</u> be named INPUT. To execute the program use the following procedure:

```
SLOPE2 / t v
/O, INPUT/PA.
/O, SPICE2/CM.
/RFL,130000.
/SPICE2.
    (system response)
/END.
    (more system response)
ALL DONE
```

The /'s in the first column are SLOPE prompts. The /PA and /CM specify file types (packed ascii and core memory) and RFL sets the field length for the run. After the ALL DONE message, file PSLOPEA contains the program output. TRIX AC can be used to view the output online and ALLOUT for sending it to a printer. Use the COL1. and LONG. options with ALLOUT.

3.2.4 Input Format

The input file for SPICE2 defines the circuit, specifies the analyses to be performed, and specifies what output is to be generated. The input is a free-format style and data fields are delimited by one or more blanks, a comma, an equal sign, or a right or left parenthesis.

Names begin with an alphabetic letter and are limited to seven characters maximum. A number may be an integer, a floating point number, or be expressed in exponential notation. A number may also be followed by one of the following alphabetic scale factors:

G = 10^9

MEG = 10^6

K = 10^3

M = 10^{-3}

U = 10^{-6}

N = 10^{-9}

P = 10^{-12}

Letters which immediately follow a scale factor are ignored as are nonscale factor letters following a number.

The first line in the input is a comment line of input. The last line in the input must be a .END. Except for the comment line and the .END line (first and last), all other input can be in any order.

There are 15 types of SPICE2 ELEMENT formats. Some of them are:

RESISTORS:	RXXXXXX N+ N- VALUE
CAPACITORS:	CXXXXXX N+ N- VALUE [IC=INCOND]
INDUCTORS:	LXXXXXX N+ N- VALUE [IC=INCOND]
COUPLED INDUCTORS:	KXXXXXX LYYYYYY LZZZZZZ VALUE
TRANSMISSION LINES:	TXXXXXX N1 N2 N3 N4 Z0 = VALUE [TD = VALUE]
INDEPENDENT VOLTAGE SOURCE:	VXXXXXX N+ N- [DC/TRVAL] [AC[ACMAG [ACPHS]]]
INDEPENDENT CURRENT SOURCE:	IXXXXXX N+ N- [DC/TRVAL] [AC[ACMAG [ACPHS]]]
LINEAR VOLTAGE-CONTROLLED CURRENT SOURCE	GXXXXXX N+ N- NC+ NC- VALUE
NONLINEAR VOLTAGE-CONTROLLED CURRENT SOURCE	NXXXXXX N+ N- [POLY(ND)] NC+ NC- P0 [P1...]
DIODE	DXXXXXX NA NC MODEL [AREA] [OFF]
BJT	QXXXXXX NC NB NE MODEL [AREA] [OFF]
JFET	JXXXXXX ND NG NS MODEL [AREA] [OFF]
MOSFET	MXXXXXX ND NG NS NB MODEL [AREA] [OFF]

The strings XXXXXX, YYYYYY, and ZZZZZZ are arbitrary alphanumeric strings. Data fields enclosed in [] are optional. Nonlinear capacitors and inductors can also be described by substitutions for the value POLY P0 P1 P2 ... where then the capacitance is expressed as a function of the voltage across the element and the inductance is a function of the current through the inductor and the values are computed as

```
VALUE = P0 + P1 * V + P2 * V ** 2 + ...  (capacitors)
VALUE = P0 + P1 * I + P2 * I ** 2 + ...  (inductors)
```

The semiconductor devices are defined on separate .MODEL input. The format for this input is:

```
.MODEL name type [parameter values]
```

Model parameter values are input through the use of keywords. Refer to the user's manual for the models, the keywords, and default parameter values.

The use of subcircuits allow elements to be defined and referenced in a similar fashion to device models. A subcircuit name begins with the letter X. The format for the invocation is:

```
Xnnnnnn node 1 node 2 ... node N  name
```

The subcircuit definition format is:

```
.SUBCKT name node1 node2 ... nodeN

                    (element interconnections)

.ENDS [name]
```

The name on the ENDS card is needed only when nested subcircuit definitions are being made.

In addition to specifying the circuit, the SPICE2 input deck specifies the analyses to be performed and the output to be printed. These options are done through the use of control cards. The SPICE2 control cards are:

```
.TEMP          .SENS
.WIDTH         .AC
.OPTIONS       .DISTO
.OP            .NOISE
.DC            .TRAN
.NODESET       .FOUR
.IC            .PRINT
.TF            .PLOT
```

The title, *comment, and .END control cards have already been covered. The .PRINT and .TRAN cards will be briefly explained below. The user manual should be consulted for the use of the other control card options.

The .PRINT specifies an output variable and the analyses in which output is to be printed or plotted. For example:

```
.PRINT  TRAN  V(4)  I(VIN)  V(8,3)
```

will print the transient voltage response for the voltage at node 4, the VIN current, and the voltage between nodes 8 and 3.

The .TRAN card contains the transient analysis simulation controls. As a minimum, the card

```
.TRAN 1NS 100NS
```

will call for a transient analysis starting at time zero to 100 ns in 1 ns steps.

3.2.5 LLL Contact

If there are any questions or comments concerning the use or operation of SPICE2, contact Waldo Magnuson, Ext. 2-9950, Bldg. 131, Rm. 2638. User's manuals are also available from Waldo.

3.2.6 References

1. "SPICEGUIDE SPICE2 User's Guide," LLL documentation, 3-3-82. (Available from OCTOPUS directory .558850:SPICE:2G.5.)

2. Laurence W. Nagel, "SPICE2: A Computer Program to Simulate Semiconductor Circuits," Electronics Research Laboratory, University of California, Berkeley, ERL-M520, May 9, 1975.

3.3 NET-2

3.3.1 Purpose and General Description

The NET-2 Network Analysis Program is a general purpose digital comuter program operational on the CDC 7600 computers at LLL under the LTSS system. The program solves the nonlinear time domain response and the linearized frequency domain response of arbitrary networks composed of electric circuit elements and system operational elements. NET-2 performs parameter variation studies, statistical studies, and network performance optimization.

NET-2 is capable of handling a variety of network components including electronics, control systems, heat flow, mechanical systems, systems of nonlinear equations, and digital logic. The program has nuclear radiation effects modeling capability for studying the effects of gamma radiation and neutrons on circuit and system response.

3.3.2 Limitations and Applicability

In test runs, NET-2 has performed well and appears to be a stable program using good numerical techniques. Good error checking is done on the input and generally input difficulties are easy to find. The program itself (FORTRAN) is poorly documented and any maintenance will be minimal. Advertised program features have produced efficient (short computer runs) and results which look correct.

3.3.3 Availability and Operation

NET-2 is available for execution on the octopus CDC 7600 computers. The program is available from public file EELIB on the R and U machines by typing:

```
X EELIB NET2 DR. / t v
```

A user's guide is available by typing:

```
X EELIB NET2GUIDE DR. / t v
```

Preliminary to executing NET2, the input file defining the network to be analyzed must be in a disk file.

The program can be executed direcily by typing:

```
NET2 / t v
```

This assumes the input data file is a disk file named INPUT. Program output will be written into file OUTPUT.

NET-2 can also be executed by typing:

```
NET2  INPUT=infilename, OUTPUT=outfilename / t v
```

where:

infilename = the name of the disk file containing the input data.
outfilename = the name of the file where the output will be written.

Use the LONG. and COL1. options if ALLOUT is used to send the output file to a printer.

3.3.4 Input Format

Input to the NET-2 program consists of a disk file description of card images in columns 1 through 72. In general, the order in which the various entries is immaterial, except for the STATE, MONTECARLO, and OPTIMIZE entries. NET-2 input is organized into a series of entries with one or more lines in each entry. The various input lines are written at specified indentation levels so that the complete input has a structured appearance or an outline form. The first line of a given entry begins in column 1 and subsequent lines of that entry are indented.

NET-2 has a rich input language permitting the following electrical circuit elements to be used in the input:

Resistor
Capacitor
Inductor
Switch
Current Source
Transmission Line
Primary Photocurrent Generator
Junction Diode
Tunnel Diode
MOSFET
Core Winding
Combinance
Diffusance
Linvill pn Junction
Voltage Controlled Conductance
Radiation Effects Capacitor
Coefficient of Coupling
Voltage Source
Voltage Controlled Voltage Source
Voltage Controlled Current Source
Nonlinear Voltage Controlled Current Source
Zener Diode
Bipolar Transistor
JFET
Magnetic Core
Storance
Driftance
Current Controlled Nodal Variable

In addition to the circuit elements, there are over 38 system elements such SUM, GAIN, MULT, MAX, DERIV, LOG, SIN, DELAY, AND, NAND, EOR, and others. System elements will not be covered in this description.

The format for circuit elements follows a standard pattern. For example, for a resistor the format is:

R (p) a b value

where R = element ID
p = optional parallel segment disignation
a and b = node names
value = resistance value

The ID consists of a prefix and a suffix. The prefix is composed of alphabetic characters only and denotes the element type (R for resistor, C for capacitor, etc.). The suffix must begin with anumeric character, followed by any combination of alphabetic and numeric characters.

Network nodes are assigned arbitrary names. Any combination of a alphabetic and numeric characters are allowed. All node voltages are measured with respect to a datum or ground node which is designated by the integer 0.

Values may be numerical constants or in symbolic form using mathematical expressions and/or symbolic constants. Symoblic constants are defined as:

Pn Value

and then can be referenced by the symbolic name. The value is a numerical constant.

Mathematical expressions use constructions similar to those used in FORTRAN. The normal arithmetic operations, use of parentheses, mathematical functions, and tables are allowed.

Besides specifying circuit elements and models, a user of NET-2 may define and use subnetworks any may define and callup stored models. Refer to the NET-2 user's manual for a discussion of SUBNETWORKS.

Three types of calculations are available in NET-2:

State solutions,
Monte Carlo solutions, and
Optimization solutions.

Initially, a user will probably be most interested in DC steady state, transient, and possibly AC response calculations. These are specified in the state entry. For example:

```
STATE1
     TIME 0 (50) 100
```

will compute the transient response from 0 to 100 in 50 steps.

```
STATE2
     TIME 0 (25) 100
     PRINT N(46)
     FREQ .005 (25*) 1.5*
     PLOT LINLOG N(OUT)
```

will calculate and print the node 46 voltage for a transient response and will plot the node OUT for a frequency calculation. Refer to the NET-2 user's manual for a description of the powerful options available in controlling output.

Comment cards can appear anywhere in an input and are specified by having an asterisk in column 1. NET-2 uses the first card in each input to label the output so it is recommended it be a comment card.

The input deck is concluded with an END card beginning in column 1.

3.3.5 LLL Contact

Like all program, a person learns most by using the program. The NET-2 user's manual should be the first source for help in preparing input. The NET2GUIDE should help for information on running NET-2 on the OCTOPUS system. Further help can be obtained from Waldo Magnuson, Ext. 2-9950, Bldg. 131, Rm. 2638.

3.3.6 References

1. "NET-2 User's Guide NET2GUIDE," LLL OCTOPUS documentation, 6-1-79. (Available from OCTOPUS directory .558850:NET2:NET2GUIDE and from EELIB).

2. A. F. Malmberg, "User's Manual, NET-2 Network Analysis Program, Release 9," Harry Diamond Laboratory document HDL-050-1, September 1973. (Available from Waldo Magnuson.)

3.4 CALAHAN

3.4.1 Purpose and General Description

The CALAHAN program was one of the first general programs written based on a topological formulation using tree-enumeration as the theoretical basis for analysis. Because of the tree-enumeration combinatorial problem (lots of trees for even modest size networks), the program has an upper limit of about 15 nodes and 30 branches. For larger network, the cost (computer time) of analyzing them becomes prohibitive.

The program can calculate the time and frequency respone of linear circuits containing both active and passive elemets. The network to be analyzed may be described either in terms of the branches (described by their node numbers, element types, and element values) or, in terms of the network function, given in polynomial form. The network is treated as a two-port network and only network functions related to the input and output ports may be requested.

3.4.2 Limitations and Applicability

The analysis time (computer time) grows exponentially with circuit size and the upper limit on sizes of circuits is 15 nodes and 30 branches. Internally, the program is dimensioned for 30 nodes, 10 passive elemets, and 20 active elements (voltage-controlled current sources) and 100 branches.

The input format is in a fixed format but is relatively simple and the input for most cirucits can be prepared in a few minutes.

Output can be any of the following:

Coefficients of the network function
Poles and Zeros of the network function
Frequency response
Time response
Repeated outputs with some element value as a parameter
Network function in symbolic form

The CALAHAN program is easy to use and can provide a quick answer for small textbook type networks. It is particularly useful if there is an interest in the poles and zeros for the respose. It should not be used for large, complex circuits and for circuits with nonlinear elements.

Some general limitations on the program are:

The nodes must be numbered consecutively 1 through N.
The network must be connected.

The RLC elements must be listed before controlled sources.
Elements across the same node pair must have nodes in the same order.
Zero valus for R and L are not permitted.
Only linear elements are accepted.

3.4.3 Availability and Operation

The CALAHAN program may be run on the OCTOPUS CDC 7600 R and U computers by typing:

```
X EELIB CALAHAN / t v
```

This extracts the program from library file EELIB and starts it into execution. The program requests the name of the file which contains the input.

A brief user's guide, CALAHANGD, is also available from OCTOPUS storage by typing:

```
XPORT RD  .558850:CALAHAN:CALAHANGD
```

3.4.4 Input Format

All input to the CALAHAN is in a fixed format so care is needed in preparing the input file. Any consistent set of units may be used, however the program output will be labeled in ohms, volts, seconds, etc. In the following abbreviated description of the input where spaces are indicated, they must exist. Field widths must be as indicated, for example, 01 is a numeric field with a width of two.

For topological input:

01 (The first card contains any positive number in columns 1 and 2 indicating that a network description follows.)

```
01 02 03 04 05 06 07 08 09 10 11
```

where

01	number of passive elements
02	number of controlled sources
03	number of nodes
04	input node

05 reference node for input

06 output node

07 reference node for output

08 key 1

where key 1 is

- 01 voltage transfer
- 02 input impedance
- 03 transfer impedance
- 04 input admittance
- 05 transfer admittance

09 key 2

where key 2 is

- 01 network fnction coefficients, poles and zeros.
- 02 network function coefficients, poles and zeros, and frequency response.
- 03 network function coefficiets, poles and zeros, and time response.
- 04 network function coefficients, poles and zeros, frequency, and time response.
- 05 network function in symoblic form.

10
11 The last two entries specify which element is to be varied and how many values it will take on. These fields may be left blank.

01 02 X VALUE-HERE

where 01 node number
02 node number
X element type

R resistor

L inductor

C capacitor

VALUE-HERE value in a 10-space field which must have a decimal point or be right justified.

There is a limit of 100 of these cards.

01 02 03 04 X VALUE-HERE

where 01 node number of current sink

02 node number of current source

03 node number of control voltage

04 node number of control voltage reference node

X element type

G voltage controlled current source

M voltage controlled current source simulating transformers (see reference 1)

VALUE-HERE as above.

There is a limit of 20 of these cards.

1 234 for frequency response information if specified by key 2

where 1 is 1 for linear frequncy is 2 for logarithmic frequency

234 number of frequency points or points per decade (right justified).

VALUE1HEREVALUE2HERE

WHERE VALUE1HERE is low frequency (not zero for log frequency)

VALUE2HERE high frequency limit

01 for time response information if called for by key 2. The number in this field specifies the number of straight line segments of input card descriptions which follow.

```
VALUE1HEREVALUE2HERE...VALUE5HERE
```

where VALUE1HERE	initial voltage value
VALUE2HERE	initial time
VALUE3HERE	final voltage value
VALUE4HERE	final time
VALUE5HERE	time increments

00 to end input file

CALAHAN can also accept polynominal transfer function input. Refer to reference 1 for the format.

3.4.5 LLL Contact

Help can be obtained from Waldo Magnuson, Ext. 2-9950, Bldg. 131, Rm. 2638. Usually errors in running CALAHAN stem from having the data in the wrong fields or requesting an analysis and then not supplying the appropriate information.

3.4.6 References

1. "CALAHAN User's Guide CALAHANGD," LLL OCTOPUS documentation, 7-1-79. (Available from OCTOPUS directory .558850:CALAHAN:CALAHANGD)
2. D. F Dawson and W. G. Magnuson, Jr., "Linear Network Analysis Computer Program (User's Manual)," UCRL-14855, June 9, 1966.
3. W. G. Magnuson, Jr., "CALAHAN: A Program for Linear Network Analysis," LEN 22114, Rev. A, (misnumbered as LEN 22144), or UCID-15052, Rev. 1, 12-17-69

3.5 EMTP

3.5.1 Purpose and General Description

The ElectroMagnetic Transients Program (EMTP) is a large FORTRAN digital computer program which was specifically designed to predict the transient performance of electric power distribution networks. The progam was developed at the Bonneville Power Administration in the late 1960's and has had many additions and extensions by BPA and others since that time. EMTP solves the ordinary differential and/or algebraic equations associated with an arbitrary interconnection of the following elements:

Lumped R L C.

Multiphase Pi-equivalents comprised of R L C elements.

Multiphase distributed-parameter transmission lines.

Nonlinear R and L.

Time-varying R.

Switches and diodes.

Voltage and current sources.

3-phase dynamic synchronous machines.

Dynamic control systems.

The EMTP progrm was designed specifically for analyzing power distribution networks (a la PG&E) and is extensively used throughout the power industry.

3.5.2 Limitations and Applicability

The EMTP program can be redimensioned to accommodate various sizes of problems. The present limitations at LLL for EMTP are:

Maximum number of nodes	50
Maximum number of branches	75
Maximum number of series Pi-circuits	300
Maximum number of sources	100

Maximum number of switches	30
Maximum number of nonlinear elements	15
Maximum number of outputs	100

EMTP has been found to be numerically stable and fast on the Cray-1 and CDC 7600 computers. Trapezoidal (second-order) implicit integration is used in solving the set of real, simultaneous, algebraic equations at each time step. A nodal-admittance formulation is used in generation of the equations.

The user manual is thick and difficult to read. Preparing input for EMTP is not easy or straightforward. But the program does analyze many circuits which have given great difficulty to other programs.

3.5.3 Availability and Operation

EMTP is available at LLNL on the Cray-1 computer. The program can be obtained from storage with the executionline:

```
XPORT RD .558850:EMTP:CRAY:EMTP / t v
```

The program can be run by typing:

```
EMTP / t v
```

This assumes the input data file is a disk file named INPUT. Program output will be written into file OUTPUT.

EMTP can also be executed by typing:

```
EMTP INPUT=infilename, OUTPUT=outfilename / t v
```

where

infilename = the name of the disk file containing the input data.
outfilename = the name of the file where the output will be written.

3.5.4 Input Format

Each EMTP problem must be fully described in the following order:

BEGIN NEW DATA CASE (first card).

Special request cards (optional).

Miscellaneous data cards.

Extensions to misc. data cards (optional).

TACS cards (optional).

Linear and nonlinear branch cards.

A blank card.

Source cards.

A blank card.

Initial condition cards (optional).

Output specification cards.

Source function specification cards (optional).

Plotting specification cards.

A blank card.

Nodes are identified by 6 (or fewer)-character alphanumeric names. The ground node is left as blanks. Networks must be connected.

Because of the many options available in the EMTP program and the intent to keep this description brief, only a minimal set of input format will be described here. The input for the optional input sections will not be covered.

Miscellaneous Data Cards

After the BEGIN NEW DATA CASE header card, two miscellaneous data cards are required. The first card has seven data entries in a 7E8.0 format. The fields (in order) are:

DELTAT = The time-step size for numerical integration in seconds.

TMAX = Termination time in seconds.

XOPT = 0 means all inductance values are in millihenries.

= f means all inductance values are interpreted as 2*pi*f*L in ohms at frequency = f.

COPT = 0 means all capacitance values are in microfarads.

= f means all capacitance values are interpreted as 2*pi*f*L in micromhos at frequence = f.

Fields 5 and 6 are for internal program checking.

FSTART = the beginning simulation time (normally zero or left blank).

The second miscellaneous data card has ten fields in the format (10I8). The fields are:

IOUT = 0 or 1 Output is printed at every time step.

= k Output is printed every K-th time step.

IPLOT = 0 or 1 Use all points for plotting.

= M Use every M-th point for plotting.

=-1 No plotting.

IDOUBL = 0 No printing of network connectivity.

= 1 Network connectivity is printed.

Fields 4 through 10 control printing of other specialized output.

<u>Linear and Nonlinear Branch Cards</u>

The network topology is next described by means of the branch description input. This information is in an 80 column card-image fixed format.

The general format is:

(I2, 4A6, 3E6.2, 35X, I1)

The first field (I2) is the variable ITYPE which designates the element type the line is describing. Some of the values for ITYPE and what they designate are the following:

ITYPE	ELEMENT TYPE
0	R, L, C, RL, RC, LC, RLC, Switch
1-10	User defined source function
11	f(t) = amplitude source

12	f(t) = ramp source
13	f(t) = ramp with linear decay source
14	f(t) = sinusoidal source
15	Surge function
16	DC converter
17	Zinc oxide surge arrester
19	Universal machine description
50-59	3-phase dynamic synchronous machine sources.
60-69	Special sources for TACS control
76	Statistics switch
91	Time-varying resistance
92	Nonlinear resistance with flashover gap
93	Nonlinear inductor
94	Dynamic SiC surge arrester
96	Pseudo-nonlinear hysteretic reactor
97	Staircase time-varying resistance
98	Pseudo-nonlinear inductor
99	Pseudo-nonlinear resistor

Fields two and three (A6, A6) are the node names (bus connections) the element is connected between. Fields four and five (A6, A6) are the node names (bus connections) of a reference branch if the element is being controlled (for example a voltage controlled switch).

The next three fields (3E6.2) are element values. For example, if we are dealing with an O type branch, then these fields are the values for R, L, and C respectively. Finally, column 80 controls output options for printing and/or plotting where:

1	for branch current
2	for branch voltage
3	for both 1 and 2
4	for branch power and energy consumption.

The use of some of the elements gets quite complex and reference to the rule book must be made until experience is gained.

Switch Cards

Several types of switches are available in EMTP, but only the time-controlled switch format will be discussed here. The format is:

(I2, 2A6, 3F10.3)

The first field is 0 (for switch). The second and third files are for the nodes (bus connections) the switch is connected between. Field four is the actual switch closing time. Field five is the time before which the opening will not be allowed. The last field defines the current margin which is used to determine when switch opening is possible.

Source Cards

Several types of sources are available. The type used is specified by the ITYPE index in columns one and two. One terminal of the source must always be grounded. The format for sources is:

(I2, A6, I2, 7E10.6)

The second field (A6) gives the node name the source is connected to, and the third field specifies if the source is a voltage source (non-negative integer or zero or blank), or a current source (a negative integer). The following fields contain amplitude, frequency, three shape controlling factors, and finally the start and stop times for the source. Again, because of the many options available, the rule book must be consulted.

Plotting Cards

A minimum of three lines are required to produce printer plots. The first card is a title and must have a "2" in column 2 followed by up to 78 characters for a plot title. The second line should contain PRINTER PLOT starting in column three.

The last input line for plotting specifies information for the plot itself. This card must begin with a "1" in column 2. Column 3 has a:

4	for node voltage plots
8	for voltage differences
9	for currents or branch energies.

Column 4 indicates the units for the time axis:

1	for degrees
2	for cycles
3	for seconds
4	for milliseconds
5	for microseconds
6	for hertz
7	for logarithm of hertz

Columns 5-7 give the time units per inch. Columns 8-11 specifies the beginning time and columns 12-15 the final time for the plot.

Finally, a blank line ends the plot specification input and a final blank line the end of input.

Comment lines may be added in the input with a C in column 1 and space in column 2. Also, blank line input may either be a totally blank input line or a line with BLANK in columns 1-5 followed by any text.

3.5.5 LLL Contact

For further information on EMTP, contact Waldo Magnuson, Ext. 2-9950, Bldg. 131, Rm. 2638.

3.5.6 References

1. "EMTP User's Manual,"Bonneville Power Administration, November 1977.

2. MFECC online program write-up "Electromagnetic Transients Program," dated 12-21-77.

4. Illustrative Examples

4.1 SCEPTRE Input Examples

The first example is a relatively simple circuit in which the input is a triangular wave current source (JIN) modeled by a table of points. The interest is in the current through the inductor which models a deflection coil in a sweep circuit.

Indenting has been used to make the sections stand out better, but SCEPTRE ignores the blanks.

```
CIRCUIT DESCRIPTION
SWEEP CIRCUIT
ELEMENTS
   R1,1-2=2000.
   R2,5-2=2000.
   R3,5-GND=22.
   RY,3-4=2.2
   RA,2-GND=200.E3
   LY,4-5=1.E-3
   CA,2-GND=330.E-6
   EA,3-GND=(5.E4*VRA)
   JIN,GND-1= TABLE 1 (TIME)
OUTPUTS
   ILY,IR1,PLOT
FUNCTIONS
   TABLE 1
   0., -.450
   15.6E-6, -.450
   31.2E-6, +.450
   46.8E-6, -.450
   62.4E-6, +.450
   78.0E-6, -.450
   100.E-6, -.450
RUNCONTROLS
   MAXIMUM PRINT POINTS = 200.
   STOPTIME = 100.E-6
END
```

The second example is a magnetron firing circuit. SCEPTRE input allows FORTRAN functions and subroutines to be input by the user. The statements in these functions must obey the rules of FORTRAN. Notice, the use of initial conditions and defined parameters.

```
CIRCUIT DESCRIPTION
   MARX BANK - MAGNETRON
ELEMENTS
   CB,1-0=425.E-9
   LB1,1-2=0.8E-6
   CS1,2-0=262.E-12
   RB,2-3=6
   LB2,3-4=0.2E--6
   LP,4-6=100.E-9
   RP,6-7=10.
   CP,7-0=1.1E-9
   LM,4-5=150.E-9
   CS2,5-0=40.E-12
   RM,5-0=EQUATION 1 (VCS2,PIO,PVO,ILM)
INITIAL CONDITIONS
   VCB=450.E3
DEFINED PARAMETERS
   PIO=12.E3
   PVO=360.E3
FUNCTIONS
   EQUATION 1 (A,B,C,D)=(SOURCE(A,B,C,D))
OUTPUTS
   VCB, ILB1,VCS1,ILB2,ILM,VRM,IRM,PLOT
RUN CONTROLS
   MAX INTEGRATION PASSES=1.E6
   MAXIMUM STEP SIZE=1.E-9
   MAXIMUM PRINT POINTSS=1000.
   STOPTIME=150.E-9
SUBPROGRAM
        FUNCTION SOURCE (A,B,C,D)
        DATA SKIP/0./
        SOURCE = 1.E6
        IF ((SKIP.EQ.0.).AND.(A.GT.(B*C))) SKIP = 1.
        IF (SKIP.EQ.0.) RETURN
        SOURCE = 30.
        RETURN
        END
END
```

The next example illustrates the use of models and their use. Notice in the second invocation of 2N999A that the defined parameter P1 has been given a new value. In this example, the change has the effect of using a different transistor alpha for T2.

```
MODEL DESCRIPTION
   MODEL 2N999A (B-E-C)
   ELEMENTS
        CE,B-E=EQUATION 1 (5.,40.,TABLE 1(VCE))
```

```
            CC,B-C=EQUATION 1 (10.,400.,TABLE 2 (VCC))
            J1,B-E=DIODE TABLE 1
            J2,B-C=DIODE TABLE 2
            J3,C-B=P1*J1
        DEFINED PARAMETERS
            P1 = .98
        OUTPUTS
            VCE,VCC,J1,PLOT 2N999A
        FUNCTIONS
            EQUATION 1 (A,B,C)=(A+B*C)
            DIODE TABLE 1
                    0,0,.3,0,.65,.05,.7,.6,.72,1.4,.73,2,.74,3.4,.77,100.,8,22
            DIODE TABLE 2
                    0,0,.58,0,.62,.4,.64,1,.66,2,.67,3,.69,7,.7,12
    CIRCUIT DESCRIPTION
        ELEMENTS
            E1,1-2=TABLE 1 (TIME)
            DERIVATIVE E1=TABLE DE1
            E2,1-4=2-
            CZ,2-3=1E3
            CX,5-6=1E3
            R1,4-3=30
            R2,3-1=20
            R3,5-1=2
            R4,4-6=240
            R5,4-7=3.3
            R6,9-1=1.8
            T1,3-5-4=MODEL 2N999A
            T2,6-1-8=MODEL 2N999A (CHANGE P1=.975)
            L1,7-8=100
            L2,9-1=900
            M,L1-L2=299.7
        OUTPUTS
            VR6,VL1,VL2,PLOT
            ICZ,ICX,IR3,IR2,IR1,J3T1,J2T1,CCT1,ICCT1
        FUNCTIONS
            TABLE 1 = 0,0,50,.5,100,.5
            TABLE DE1=0,.01,50,.01,50,0,100,0
        RUN CONTROLS
            STOP TIME=500
            MAXIMUM PRINT POINTS=3000
            RUN INITIAL CONDIDTIONS
    END
```

Parameter values may also be changed over a rage of values. In the following simple RLC network, the analysis is performed for three different values of the capacitor.

```
CIRCUIT DESCRIPTION
    EXAMPLE TO ILLUSTRATE RERUN FEATURE OF SCEPTRE
ELEMENTS
    C1,GND-1=.1E6
    L1,1-2=1.
    R1,2-GND=1.E-3
INITIAL CONDITIONS
    VC1=-4800
OUTPUTS
    IR1,PLOT
RUNCONTROLS
    MINIMUM STEP SIZE = 1.E-6
    STOP TIME = 2.5E3
RERUN DESCRIPTION (3)
ELEMENTS
    C1 = 3.7E6, 4.1E6, 4.5E6
END
```

4.2 SPICE2 Input Examples

The first SPICE2 example is identical to the first SCEPTRE example. Spaces are important to SPICE2 and are used as field deliminators.

```
SWEEP CIRCUIT
R1 1 2 2000.
R2 5 2 2000.
R3 5 0 22.
RY 3 4 2.2
RA 2 0 330.E-6
EA 0 3 2 0 5.E4
IIN 0 1 PWL(0. -.45 15.6E.6 -.45 31.2E-6 +.45 46.8E-6 -.45
  + 62.RE-6 +.45 78.E-6 -.45 100.E-6 -.45)
.PLOT TRAN V(2,1) V(4,5)
.TRAN 1.E-6 100.E-6
.END
```

SPICE2 has several models built into the program. Model parameter values are supplied with the input.

```
SIMPLE RTL INVERTER
VCC 4 0 DC 5
VIN 1 0 PULSE 0 5 2NS 2NS 2NS 30NS
RP 1 2 10K
Q1 3 2 0 Q1
RC 4 3 1K
.PRINT DC V(2) V(3) I(RC)
.PLOT TRAN V(3)
.MODEL Q1 NPN BF=20 RB=100 TF=.1NS CJC=2PF
.DC VIN 0 5 0.1
.TRAN 1NS 100NS 25NS
.END
```

The next example uses the subcircuit option of SPICE2. The subcircuit models an operational amplifier (OP-AMP) with a voltage ratio of 10^5.

```
LOW PASS ACTIVE FILTER
VIN 1 0 AC 1
R1 1 2 1K
R2 2 3 1K
R3 4 0 10K
R4 5 4 95K
R5 5 0 90K
C1 3 0 .4U
C2 5 2 .06U
X1 3 4 5 0 OPAMP
.SUBCKT OPAMP 1 2 3 4
```

```
R1 1 2 150K
R2 5 3 150
C1 1 2 10P
E1 5 4 1 2 100K
.ENDS OPAMP
.PLOT AC VM(5) VDB(5)
.AC DEC 10 1HZ 100MEGHZ
.END
```

One of the models in SPICE2 is a simple transmission line model. This example again uses the subcircuit model with two transmission lines imbedded within the subcircuit.

```
TRANSMISISON-LINE INVERTER
V1 1 0 PULSE(0 1 0 0.1N)
R1 1 2 50
X1 2 0 0 4 TLINE
R2 4 0 50
.SUBCKT TLINE 1 2 3 4
T1 1 2 3 4 Z0=50 TD=1.5NS
T2 2 0 4 0 Z0=100 TD=1NS
.ENDS TLINE
.TRAN 0.1NS 20NS
.PLOT TRAN V(2) V(4)
.END
```

4.3 NET-2 Input Examples

The first example again is the same as the first example for the previous two programs but expressed in NET-2 input format. Spaces are used to delimit the fields on input lines. Indenting is also used in NET-2 input.

```
*SWEEP CIRCUIT
R1 1 2 2000
R2 5 2 2000
R3 5 0 22.
R4 3 4 2.2
R5 2 0 200.E3
L1 4 5 1.E3
C1 2 0 330.E-6
VCVS1 0 3 2 0 5.E4
I1 0 1 TABLE1 (TIME)
TABLE1
     0 -.45
     15.6E-6 -.45
     31.2E-6 +.45
     46.8E-6 -.45
     62.4E-6 +.45
     78.0E-6 -.45
     100.E-6 -.45
STATE1
     TIME 0 (100) 100.E-6
     PLOT I(L1) I(R1)
     PRINT I(L1) I(R1)
END
```

NET-2 allows a wide variety of system models to be included as input. The following example gives an idea of some of the elements available.

```
*NONLINEAR EQUATION SYSTEM
INT1 DDV DV 1
INT2 DV V 1, 7
GAIN 1 DV 1 2/(TIME+1E-6)
SUM1 DDV V -1 3
INT3 DF F 1, 1.8
MULT1 FSQ F F
SUM2 2 FSQ GSQ
INT4 DG G 1
MULT2 GSQ G G
SUM3 4 -V P2
MULT3 DF 4 G
GAIN3 G 7 2/(TIME+1E-6)
SUM4 5 V P1
MULT6 6 F 5
SUM5 DG 6 -7
```

```
V1 P1 0 15
V2 P2 0 1
GAIN2 2 3 -2
STATE1
     TIME 0 (100) 2
     PLOT N(V)
     PLOT N(G)
     PLOT N(F)
END
```

There are several complex models for active and passive built into NET-2. The following two examples illustrate how parameter information is input to built-in models.

```
MB TRANSISTOR AMPLIFIER DESIGN
*UNITS- V, MA, KOHMS, PF, UH, NS, GHZ
LIBRARY

    T    2N918  4
            GE  2.E-5
            GC  2.E-5
            BN  42.5
            BI  .97
            IES 2.58E-12
            ICS 2.12E-12
            THE 38
            THC 36.3
            CE  2.
            NE  .5
            CC  1.7
            NC  .8
            RBB .012
            RCC .0115
V1  7    0       -9
R1  0    2       12
R2  2    5       12
R3  3    5       2.7
R4  1    0       .47
R5  6    0       .050
C1  1    2       470
C2  4    6       12
C3  6    0       30
C4  5    0       .01E6
C5  3    0       .01E6
L1  4    0       .39
L2  7    5       8.2
STATE1
    TIME 0
    PRINT N (2) N (3)    N (4)
STATE2
    FREQ    025     (50*)       .125
    PLOT    LINLOG  A(6-0/1-0) VS FREQ
    PRINT   A (6-0/1-0) VSA FREQ
END
```

Magnetic Core windings are allowed on cores in which the magnetic properties of the core material may be input. The following example models a ferrite switching core with two windings.

```
TRANSFORMER MODEL
LIBRARY
     MC   CORE1  9
          OD     0.4
          ID     0.2
          PHIR   .03
          TYPE   2
          FD1    20
          PHID2  -.01
          HA     1000
          HQ     20
          HN     10
          FOPP   300
          FB     1000
          FO     500
          RCP    1

V1  1    0      x1
X1 = SIN (125664*TIME)
CW1  1  0  MC1  24
CW2  3  4  MC1  480
MC1      CORE1
STATE1
    TIME 0      (100)    50E-6
    PLOT N(1)   VS TIME
    PLOT V(CW2) VS TIME
    PLOT V(CW1) VS TIME
END
```

-2 has both a lumped parameter and dispersive transmission line model in program.

```
NET-2 TRANSMISSION LINE ANALYSIS
C1 0 6 43
V1 6 1 -2E5*U(TIME-.1)
R1 1 2 .03
C2 2 3 200
C3 4 5 200
R2 5 0 .03
L1 4 0 . 105
TLINE1 3 4 0 10 .017 26
MAXSTEP = 1
TERMINATE = 100
STATE1
     TIME 0 (200) 100
     PLOT V(C1)
END
```

4.4 CALAHAN Input Examples

CALAHAN is simple to use but because of the fixed format for input, is not very forgiving. This example is a time-domain analysis of a feedback amplifier.

```
01 FEEDBACK AMPLIFIER
18 04 07 02 01 01 07 01 03 18 04
02 03 R 1.0
03 01 R 1.0
03 01 C 100.0
03 04 R 50.0
04 01 R 1.0
04 01 C 5.0
05 04 C 5.0
05 01 R 50.0
05 06 R 1.0
05 06 C 95.0
01 05 C 5.0
01 06 R 50.0
06 07 R 1.0
06 07 C 95.0
01 06 C 5.0
01 07 R 50.0
07 01 R 0.1
07 03 R 0.5
04 03 01 03 G 100.0
05 01 04 01 G 100.0
01 06 05 06 G 100.0
01 07 06 07 G 100.0
05      1.0     2.0    5.0
01
1.0     0.0     1.0          74.0     0.5
00
```

Frequency-domain analyses are also easily specified with CALAHAN. In the following simple circuit, phase and magnitude are specified for output.

```
01 SIMPLE INVERTER
05 01 05 01 05 03 05 01 02
01 02 R 1.0
02 04 R 1.0
04 05 R 2000.
03 05 R .075
04 05 C .16
05 03 04 05 G 667000.
2 010
.001    100.
00
```

4.5 EMTP Input Examples

EMTP input is awkward. No suitable user manual[1] exists. The "Rule Book" published by Bonneville Power Administration has a bare minimum of user instructions. Rather, it is more of a reference book related to EMTP input and output. The following two examples should convey a flavor of the EMTP input format.

```
BEGIN NEW DATA CASE
C   TEST PROBLEM HAVING ONE SINUSOIDAL SOURCE,
C   ONE RESISTOR, ONE SWITCH, AND ONE SINGLE-
C   PHASE P1-CIRCUIT.
    .05         1.0      1.0    1.0
    1           1        1      1        1     -1
    2           -1
    rec                  .001
    1 SEND REC           1.0    1.0      1.E4
BLANK CARD ENDING BRANCH CARDS
    GEN  SEND            -1.0   1.0
BLANK CARD ENDING SWITCH CARDS
14 GEN                   100    1.0
BLANK CARD ENDING SOURCE CARDS
    1
         PRINTER PLOT
193 .2   .0     1.0 REC
BLANK CARD ENDING PLOT CARDS
BLANK CARD ENDING INPUT
BEGIN NEW DATA ASE
    1.e3 100.-3
         1      1     1     1     1     0     -1
    TRANSFORMER                       X-FORM
         210.0        1050.0
          9999
    1NODE06NODE26              0.   15.49  12500
    2NODE27NODE22              0.   637.0 80400
    TRANSFORMER                       X-FORM
         210.0        1050.0
          9999
    1NODE26NODE36              0   15.49  12500
    2NODE27NODE22              0.   637.  80400
    TRANSFORMER                       X-FORM
         210.0        1050.0
          9999
    1NODE36NODE06              0.   15.49  12500
    2NODE37NODE22              0.   637.   80400
    0NODE02NODE08     225.0         0.     .250
    0NODE03NODE08     225.0         0.     .250
    0NODE04NODE09     132.0         0.     0.68
```

```
ONODE24NODE09     132.0        0.    0.68
ONODE34NODE09     132.0        0.    0.68
ONODE05NODE10     360.0        0.    .375
ONODE25NODE10     360.0        0.    .375
ONODE35NODE10     360.0        0.    .375
ONODE06NODE11     360.0        0.    .375
ONODE26NODE11     360.0        0.    .375
ONODE36NODE11     360.0        0.    .375
ONODE07NODE12  17000.          0.  .00925
ONODE27NODE12  17000.          0.  .00925
ONODE37NODE12  17000..         0.  .00925
ONODE20           878.0        0.     3.5
ONODE01NODE04       .36        0.      0.
ONODE02NODE24       .36        0.      0.
ONODE03NODE34       .36        0.      0.
ONODE04NODE05        0.        6.0.    0.
ONODE24NODE25        0.        6.0.    0.
ONODE34NODE35        0.        6.0.    0.
ONODE07NODE20  23.0E6          0.    .375
ONODE27NODE20  23.0E6          0.    .375
ONODE37NODE20  23.0E6          0.    .375
ONODE07NODE20        0.        0. .287E-6
ONODE27NODE20        0.        0. .287E-6
ONODE37NODE20        0.        0. .287E-6
ONODE07        23.0E6          0.      0.
ONODE27        23.0E6          0.      0.
ONODE37        23.0E6.         0.      0.
ONODE07             0.         0.  287E-6
ONODE27             0.         0.  287E-6
ONODE37             0.         0.  286E-6
ONODE21           2169.
ONODE11             1.E6
ONODE18NODE20       1.0        0.    0.
ONODE19NODE07       1.0        0.    0.
ONODE28NODE20       1.0        0.    0.
ONODE29NODE27       1.0        0.    0.
ONODE38NODE20       1.0        0.    0.
ONODE39NODE37       1.0        0.    0.
(blank card)
ONODE05NODE06 -1.0   .050          5.0
ONODE25NODE26 -1.0   .050          5.0
ONODE35NODE36 -1.0   .050          5.0
ONODE20NODE21 .040   -1.0
11NODE07NODE18
11NODE27NODE28
11NODE37NODE38
11       NODE19
11       NODE29
```

```
11      NODE39
 (blank card)
14NODE01   10210.    60.0         0.0                -1.0
14NODE02   10210.    60.0       120.0                -1.0
14NODE03   10210.    60.0       240.0                -1.0
 (blank card)
```