- Parent Directory -

INTERACTIVE INTERPRETER

Spice3 consists of a simulator and a front-end for data analysis and plotting. The front-end may be run as a separate "stand-alone" program under the name **Nutmeg**.

Nutmeg will read in the "raw" data output file created by **spice -r** or with the **write** command in an interactive Spice3 session. Nutmeg or interactive Spice3 can plot data from a simulation on a graphics terminal or a workstation display. Most of the commands available in the interactive Spice3 front end are available in **nutmeg**; where this is not the case, Spice-only commands have been marked with an asterisk ("*"). Note that the raw output file is different from the data that Spice2 writes to the standard output, which may also be produced by spice3 with the "-b" command line option.

Spice and **Nutmeg** use the **X Window System** for plotting if they find the environment variable **DISPLAY**. Otherwise, a graphics-terminal independent interface (**MFB**) is used. If you are using **X** on a workstation, the **DISPLAY** variable should already be set; if you want to display graphics on a system different from the one you are running **Spice3** or **Nutmeg** on, **DISPLAY** should be of the form "*machine*:0.0". See the appropriate documentation on the X Window Sytem for more details.

Command Synopsis

```
spice [ -n ] [ -t term ] [ -r rawfile] [ -b ] [ -i ] [ input
file ... ]

nutmeg [ - ] [ -n ] [ -t term ] [ datafile ... ]
```

Options are:

- **-** Don't try to load the default data file ("rawspice.raw") if no other files are given. **Nutmeg** only.

- **-n** (or **-N**)

  Don't try to source the file ".spiceinit" upon startup. Normally **spice** and **nutmeg** try to find the file in the current directory, and if it is not found then in the user's home directory.

- **-t term** (or **-T term**)

  The program is being run on a terminal with *mfb* name **term**.

- **-b** (or **-B**)

  Run in batch mode. Spice3 reads the default input source (e.g. keyboard) or reads the given input file and performs the analyses specified; output is either Spice2-like line-printer plots ("ascii plots") or a spice rawfile. See the following section for details. Note that if the input source is not a terminal (e.g. using the IO redirection notation of "<") Spice3 defaults to batch mode (**-i** overrides). This option is valid for **Spice3** only.

- **-s** (or **-S**)

  Run in server mode. This is like batch mode, except that a temporary rawfile is used and then written to the standard output, preceded by a line with a single "@", after the simulation is done. This mode is used by the spice daemon. This option is valid for **Spice3** only.

- **-i** (or **-I**)

  Run in interactive mode. This is useful if the standard input is not a terminal but interactive mode is desired. Command completion is not available unless the standard input is a terminal, however. This option is valid for **Spice3** only.

- **-r** *rawfile* (or **-P** *rawfile*)

  Use *rawfile* as the default file into which the results of the simulation are saved. This option is valid for **Spice3** only.

Further arguments to **spice** are taken to be Spice3 input files, which are read and saved (if running in batch mode then they are run immediately). Spice3 accepts most Spice2 input file, and output ascii plots, fourier analyses, and node printouts as specified in .plot, .four, and .print cards. If an **out** parameter is given on a .width card, the effect is the same as **set width = ...**. Since Spice3 ascii plots do not use multiple ranges, however, if vectors together on a .plot card have different ranges they are not provide as much information as they would in Spice2. The output of Spice3 is also much less verbose than Spice2, in that the only data printed is that requested by the above cards.

For **nutmeg**, further arguments are taken to be data files in binary or ascii format (see **sconvert** (1)) which are loaded into nutmeg. If the file is in binary format, it may be only partially completed (useful for examining Spice2 output before the simulation is finished). One file may contain any number of data sets from different analyses.

EXPRESSIONS, FUNCTIONS, AND CONSTANTS

Spice and **Nutmeg** data is in the form of vectors: time, voltage, etc. Each vector has a type, and vectors can be operated on and combined algebraicly in ways consistent with their types. Vectors are normally created when a data file is read in (see the **load** command below), and when the initial datafile is loaded. They can also be created with the **let** command.

An expression is an algebraic formula involving vectors and scalars (a scalar is a vector of length 1) and the following operations:

```
+ - * / ^ %
```

% is the modulo operator, and the comma operator has two meanings: if it is present in the argument list of a user-definable function, it serves to separate the arguments. Otherwise, the term x , y is synonymous with x + j(y).

Also available are the logical operations & (and), | (or), ! (not), and the relational operations <, >, >=, <=, =, and <> (not equal). If used in an algebraic expression they work like they would in C, producing values of 0 or 1. The relational operators have the following synonyms:

```
gt >
lt <
ge >=
```

```
le <=
ne <>
eq \\&=
and &
or |
not !
```

These are useful when < and > might be confused with IO redirection (which is almost always).

The following functions are available:

mag(vector)	The magnitude of vector
ph(vector)	The phase of vector
j(vector)	i (sqrt(-1)) times vector
real(vector)	The real component of vector
imag(vector)	The imaginary part of vector
db(vector)	20 log10(mag(vector))
log(vector)	The logarithm (base 10) of vector
ln(vector)	The natural logarithm (base e) of vector
exp(vector)	e to the vector power
abs(vector)	The absolute value of vector.
sqrt(vector)	The square root of vector.
sin(vector)	The sine of vector.
cos(vector)	The cosine of vector.
tan(vector)	The tangent of vector.
atan(vector)	The inverse tangent of vector.
norm(vector)	The **vector** normalized to 1(i.e,the largest magnitude of any componentis 1)
rnd(vector)	A vector with each component a random integer between 0 and the absolute value of the vectors's corresponding component.
mean(vector)	The result is a scalar (a length 1 vector)that is the mean of the elements of **vector**.
vector(number)	The result is a vector of length **number**, with elements 0, 1, ... **number - 1**. If **number** is a vector then just the first element is taken, and if it isn't an integer then the floor of the magnitude is used.
lenght(vector)	The length of **vector**.
interpolate (plot.vector)	The result of interpolating the named vector onto the scale of the current plot. This function uses the variable **polydegree** to determine the degree of interpolation.
deriv(vector)	Calculates the derivative of the given vector. This uses numeric differentiation by interpolating a polynomial and may not produce satisfactory results (particularly with iterated differentiation).The implementation only caculates the dirivative with respect to the real componant of that vector's scale.

A vector may be either the name of a vector already defined or a floating-point number (a scalar). A number may be written in any format acceptable to SPICE, such as **14.6Meg** or **-1.231e-4**. Note

that you can either use scientific notation or one of the abbreviations like *MEG* or *G*, but not both. As with SPICE, a number may have trailing alphabetic characters after it.

The notation **expr [num]** denotes the **num**'th element of **expr**. For multi-dimensional vectors, a vector of one less dimension is returned. Also for multi-dimensional vectors, the notation **expr[m] [n]** will return the *n*th element of the **m**th subvector. To get a subrange of a vector, use the form **expr[lower, upper]**.

To reference vectors in a plot that is not the *current plot* (see the **setplot** command, below), the notation **plotname.vecname** can be used.

Either a plotname or a vector name may be the wildcard **all**. If the plotname is **all**, matching vectors from all plots are specified, and if the vector name is **all**, all vectors in the specified plots are referenced. Note that you may not use binary operations on expressions involving wildcards it is not obvious what **all + all** should denote, for instance. Thus some (contrived) examples of expressions are:

```
cos(TIME) + db(v(3))
sin(cos(log([1 2 3 4 5 6 7 8 9 10])))
TIME * rnd(v(9)) - 15 * cos(vin#branch) ^ [7.9e5 8]
not ((ac3.FREQ[32] & tran1.TIME[10]) gt 3)
```

Vector names in **spice** may have a name such as **@name[param]**, where **name** is either the name of a device instance or model. This denotes the value of the **param** parameter of the device or model. See Appendix B for details of what parameters are available. The value is a vector of length 1. This function is also available with the **show** command, and is available with variables for convenience for command scripts.

There are a number of pre-defined constants in **nutmeg**. They are:

pi	□(3.14159...)
e	The base of natural logarithms (2.71828...)
c	The speed of light (299,792,500 m/sec)
i	The square root of -1
kelvin	Absolute 0 in Centigrade (-273.15 □C)
echarge	The charge on an electron (1.6021918e-19 C)
boltz	Boltzman's constant (1.3806226e-23)
planck	Planck's constant (h = 6.626200e-34)

These are all in MKS units. If you have another variable with a name that conflicts with one of these then it takes precedence.

COMMAND INTERPRETATION

If a word is typed as a command, and there is no built-in command with that name, the directories in the *sourcepath* list are searched in order for the file. If it is found, it is read in as a command file (as if it were **source**d). Before it is read, however, the variables *argc* and *argv* are set to the number of words following the filename on the command line, and a list of those words respectively. After the file is finished, these variables are **unset**. Note that if a command file calls another, it must save its *argv* and *argc* since they are altered. Also, command files may not be re-entrant since there are no local variables. (Of course, the procedures may explicitly manipulate a stack...) This way one can write scripts analogous to shell scripts for **nutmeg** and Spice3.

Note that for the script to work with Spice3, it must begin with a blank line (or whatever else, since it is thrown away) and then a line with **.control** on it. This is an unfortunate result of the **source** command being used for both circuit input and command file execution. Note also that this allows the user to merely type the name of a circuit file as a command and it is automatically run. The commands are executed immediately, without running any analyses that may be spicified in the circuit (to execute the analyses before the script executes, include a "run" command in the script).

There are various command scripts installed in */usr/local/lib/spice/scripts* (or whatever the path is on your machine), and the default *sourcepath* includes this directory, so you can use these command files (almost) like builtin commands.

COMMANDS

Ac*: Perform an AC, small-signal frequency response analysis

General Form

```
ac ( DEC | OCT | LIN ) N Fstart Fstop
```

Do an ac analysis. See the previous sections of this manual for more details.

Alias: Create an alias for a command

General Form

```
alias [word] [text ...]
```

Causes **word** to be aliased to **text**. History substitutions may be used, as in C-shell aliases.

Alter*: Change a device or model parameter

General Form

```
alter device value alter device parameter value [ parameter value ]
```

Alter changes the value for a device or a specified parameter of a device or model. The first form is used by simple devices which have one principal value (resistors, capacitors, etc.) where the second form is for more complex devices (bjt's, etc.). Model parameters can be changed with the second form if the name contains a "#".

For specifying vectors as values, start the vector with "[", followed by the values in the vector, and end with "]". Be sure to place a space between each of the values and before and after the "[" and "]".

Asciiplot: Plot values using old-style character plots

General Form

```
asciiplot plotargs
```

Produce a line printer plot of the vectors. The plot is sent to the standard output, so you can put it into a file with *asciiplot args ... > file*. The **set** options **width, height,** and **nobreak** determine the width and height of the plot, and whether there are page breaks, respectively. Note that you will have problems if you try to **asciiplot** something with an X-scale that isn't monotonic (i.e, something like *sin(TIME)*), because **asciiplot** uses a simple-minded linear interpolation.

Aspice: Asynchronous spice run

General Form

```
aspice input-file [output-file]
```

Start a SPICE-3 run, and when it is finished load the resulting data. The raw data is kept in a temporary file. If *output-file* is specified then the diagnostic output is directed into that file, otherwise it is thrown away.

Bug: Mail a bug report

General Form

```
bug
```

Send a bug report. Please include a short summary of the problem, the version number and name of the operating system that you are running, the version of Spice that you are running, and the relevant spice input file. (If you have defined BUGADDR, the mail is delivered to there.)

Cd: Change directory

General Form

```
cd [directory]
```

Change the current working directory to **directory**, or to the user's home directory if none is given.

Destroy: Delete a data set

General Form

```
destroy [plotnames | all]
```

Release the memory holding the data for the specified runs.

Dc*: Perform a DC-sweep analysis

General Form

```
dc Source-Name Vstart Vstop Vincr [ Source2 Vstart2 Vstop2 Vincr2 ]
```

Do a dc transfer curve analysis. See the previous sections of this manual for more details.

Define: Define a function

General Form

```
define function(arg1, arg2, ...) expression
```

Define the *user-definable function* with the name *function* and arguments *arg1, arg2, ...* to be *expression*, which may involve the arguments. When the function is later used, the arguments it is given are substituted for the formal arguments when it is parsed. If *expression* is not present, any definition for *function* is printed, and if there are no arguments to *define* then all currently active definitions are printed. Note that you may have different functions defined with the same name but different arities.

Some useful definitions are:

```
define max(x,y) (x > y) * x + (x <= y) * y define min(x,y) (x < y) *
x + (x >= y) * y
```

Delete*: Remove a trace or breakpoint

General Form

```
delete [ debug-number ... ]
```

Delete the specified breakpoints and traces. The **debug numbers** are those shown by the **status** command (unless you do **status > file**, in which case the debug numbers are not printed).

Diff: Compare vectors

General Form

```
diff plot1 plot2 [vec ...]
```

Compare all the vectors in the specified *plots*, or only the named vectors if any are given. There are different vectors in the two plots, or any values in the vectors differ significantly the difference is reported. The variable **diff_abstol**, **diff_reltol,** and **diff_vntol** are used to determine a significant difference.

Display: List known vectors and types

General Form

```
display [varname ...]
```

Prints a summary of currently defined vectors, or of the names specified. The vectors are sorted by name unless the variable **nosort** is set. The information given is the name of the vector, the length, the type of the vector, and whether it is real or complex data. Additionally, one vector is labeled **[scale]**. When a command such as *plot* is given without a *vs* argument, this scale is used for the X-axis. It is always the first vector in a rawfile, or the first vector defined in a new plot. If you undefine the scale (i.e, *let TIME = []*), one of the remaining vectors becomes the new scale (which is undetermined).

Echo: Print text

General Form

```
echo [text...]
```

Echos the given text to the screen.

Edit*: Edit the current circuit

General Form

```
edit [ file ]
```

Print the current Spice3 input file into a file, call up the editor on that file and allow the user to modify it, and then read it back in, replacing the original file. If a *filename* is given, then edit that file and load it, making the circuit the current one.

Fourier: Perform a fourier transform

General Form

```
fourier fundamental_frequency [value ...]
```

Does a fourier analysis of each of the given values, using the first 10 multiples of the fundamental frequency (or the first *nfreqs*, if that variable is set see below). The output is like that of the **.four** Spice3 line. The values may be any valid expression. The values are interpolated onto a fixed-space grid with the number of points given by the **fourgridsize** variable, or 200 if it is not set. The interpolation is of degree **polydegree** if that variable is set, or 1. If **polydegree** is 0, then no interpolation is done. This is likely to give erroneous results if the time scale is not monotonic, though.

Hardcopy: Save a plot to a file for printing

General Form

```
hardcopy file plotargs
```

Just like **plot**, except creates a file called **file** containing the plot. The file is an image in *plot(5)* format, and can be printed by either the **plot(1)** program or **lpr** with the **-g** flag.

Help: Print summaries of Spice3 commands

General Form

```
help [all] [command ...]
```

Prints help. If the argument **all** is given, a short description of everything you could possibly type is printed. If **command**s are given, descriptions of those commands are printed. Otherwise help for only a few major commands is printed.

History: Review previous commands

General Form

```
history [number]
```

Print out the history, or the last **number** commands typed at the keyboard. *Note:* in Spice3 version 3a7 and earlier, all commands (including ones read from files) were saved.

Iplot*: Incremental plot

General Form

```
iplot [ node ...]
```

Incrementally plot the values of the nodes while Spice3 runs. The **iplot** command can be used with the **where** command to find trouble spots in a transient simulation.

Jobs: List active asynchronous spice runs

General Form

```
jobs
```

Report on the asynchronous SPICE-3 jobs currently running. **Nutmeg** checks to see if the jobs are finished every time you execute a command. If it is done then the data is loaded and becomes available.

Let: Assign a value to a vector

General Form

```
let name = expr
```

Creates a new vector called **name** with the value specified by **expr,** an expression as described above. If **expr** is [] (a zero-length vector) then the vector becomes undefined. Individual elements of a vector may be modified by appending a subscript to **name** (ex. **name[0]**). If there are no arguments, **let** is the same as **display**.

Linearize*: Interpolate to a linear scale

General Form

```
linearize vec ...
```

Create a new plot with all of the vectors in the current plot, or only those mentioned if arguments are given. The new vectors are interpolated onto a linear time scale, which is determined by the values of **tstep, tstart,** and **tstop** in the currently active transient analysis. The currently loaded input file must include a transient analysis (a **tran** command may be run interactively before the last **reset**, alternately), and the current plot must be from this transient analysis. This command is needed because Spice3 doesn't output the results from a transient analysis in the same manner that Spice2 did.

Listing*: Print a listing of the current circuit

General Form

```
listing [logical] [physical] [deck] [expand]
```

If the **logical** argument is given, the listing is with all continuation lines collapsed into one line, and if the **physical** argument is given the lines are printed out as they were found in the file. The default is **logical**. A **deck** listing is just like the **physical** listing, except without the line numbers it recreates the input file verbatim (except that it does not preserve case). If the word **expand** is present, the circuit is printed with all subcircuits expanded.

Load: Load rawfile data

General Form

```
load [filename] ...
```

Loads either binary or ascii format rawfile data from the files named. The default filename is **rawspice.raw**, or the argument to the **-r** flag if there was one.

Op*: Perform an operating point analysis

General Form

```
op
```

Do an operating point analysis. See the previous sections of this manual for more details.

Plot: Plot values on the display

General Form

```
plot exprs [ylimit ylo yhi] [xlimit xlo xhi] [xindices xilo xihi]
[xcompress comp] [xdelta xdel] [ydelta ydel] [xlog] [ylog] [loglog] [vs
xname] [xlabel word] [ylabel word] [title word] [samep] [linear]
```

Plot the given **exprs** on the screen (if you are on a graphics terminal). The **xlimit** and **ylimit** arguments determine the high and low x- and y-limits of the axes, respectively. The **xindices** arguments determine what range of points are to be plotted everything between the **xilo**'th point and the **xihi**'th point is plotted. The **xcompress** argument specifies that only one out of every **comp** points should be plotted. If an **xdelta** or a **ydelta** parameter is present, it specifies the spacing between grid lines on the X- and Y-axis. These parameter names may be abbreviated to **xl, yl, xind, xcomp, xdel,** and **ydel** respectively.

The **xname** argument is an expression to use as the scale on the x-axis. If **xlog** or **ylog** are present then the X or Y scale, respectively, is logarithmic (**loglog** is the same as specifying both). The **xlabel** and **ylabel** arguments cause the specified labels to be used for the X and Y axes, respectively.

If **samep** is given, the values of the other parameters (other than **xname**) from the previous **plot, hardcopy,** or **asciiplot** command is used unless re-defined on the command line.

The **title** argument is used in the place of the plot name at the bottom of the graph.

The **linear** keyword is used to override a default log-scale plot (as in the output for an AC analysis).

Finally, the keyword **polar** to generate a polar plot. To produce a smith plot, use the keyword **smith**. Note that the data is transformed, so for smith plots you will see the data transformed by the function (x - 1) / (x + 1). To produce a polar plot with a smith grid but without performing the smith transform, use the keyword **smithgrid**.

Print: Print values

General Form

```
print [col] [line] expr ...
```

Prints the vector described by the expression **expr.** If the **col** argument is present, print the vectors named side by side. If **line** is given, the vectors are printed horizontally. **col** is the default, unless all the vectors named have a length of one, in which case **line** is the default. The options **width, length,** and **nobreak** are effective for this command (see **asciiplot**). If the expression is **all**, all of the vectors available are printed. Thus **print col all > file** prints everything in the file in SPICE2 format. The scale vector (time, frequency) is always in the first column unless the variable **noprintscale** is true.

Quit: Leave Spice3 or Nutmeg

General Form

```
quit
```

Quit nutmeg or spice.

Rehash: Reset internal hash tables

General Form

```
rehash
```

Recalculate the internal hash tables used when looking up UNIX commands, and make all UNIX commands in the user's PATH available for command completion. This is useless unless you have **set unixcom** first (see above).

Reset*: Reset an analysis

General Form

```
reset
```

Throw out any intermediate data in the circuit (e.g, after a breakpoint or after one or more analyses have been done already), and re-parse the input file. The circuit can then be re-run from it's initial state, overriding the affect of any **set** or **alter** commands. In Spice-3e and earlier versions this was done automatically by the **run** command.

Reshape: Alter the dimensionality or dimensions of a vector

General Form

```
reshape vector vector ... or reshape vector vector ... [ dimension,
dimension, ... ] or reshape vector vector ... [ dimension ]
[ dimension ] ...
```

This command changes the dimensions of a vector or a set of vectors. The final dimension may be left off and it will be filled in automatically. If no dimensions are specified, then the dimensions of the first vector are copied to the other vectors. An error message of the form 'dimensions of *x* were inconsistent' can be ignored.

Resume*: Continue a simulation after a stop

General Form

```
resume
```

Resume a simulation after a stop or interruption (control-C).

Rspice: Remote spice submission

General Form

```
rspice input file
```

Runs a SPICE-3 remotely taking the **input file** as a SPICE-3 input file, or the current circuit if no argument is given. **Nutmeg** or **Spice3** waits for the job to complete, and passes output from the remote job to the user's standard output. When the job is finished the data is loaded in as with aspice. If the variable *rhost* is set, **nutmeg** connects to this host instead of the default remote Spice3 server machine. This command uses the "rsh" command and thereby requires authentication via a ".rhosts" file or other equivalent method. Note that "rsh" refers to the "remote shell" program, which may be "remsh" on your system; to override the default name of "rsh", set the variable *remote_shell*. If the variable *rprogram* is set, then **rspice** uses this as the pathname to the program to run on the remote system.

Note: rspice will not acknowledge elements that have been changed via the "alter" or "altermod" commands.

Run*: Run analysis from the input file

General Form

```
run [rawfile]
```

Run the simulation as specified in the input file. If there were any of the control lines .ac, .op, .tran, or .dc, they are executed. The output is put in **rawfile** if it was given, in addition to being available interactively. In Spice-3e and earlier versions, the input file would be re-read and any affects of the **set** or **alter** commands would be reversed. This is no longer the affect.

Rusage: Resource usage

General Form

```
rusage [resource ...]
```

Print resource usage statistics. If any **resource**s are given, just print the usage of that resource. Most resources require that a circuit be loaded. Currently valid **resource**s are:

elapse	The amount of time elapsed since the last **rusage elapsed** call
faults	Number of page faults and context switches (BSD only).
space	Data space used.
time	CPU time used so far.

temp	Operating temperature.
tnom	Temperature at which device parameters were measured.
equations	Circuit Equations

time	Total Analysis Time
totiter	Total iterations
accept	Accepted timepoints
rejected	Rejected timepoints

loadtime	Time spent loading the circuit matrix and RHS
reordertime	Matrix reordering time
lutime	L-U decomposition time
solvetime	Matrix solve time

trantime	Transient analysis time
tranpoints	Transient timepoints
traniter	Transient iterations
trancuriters	Transient iterations for the last time point*
tranlutime	Transient L-U decomposition time
transolvetime	Transient matrix solve time

everything	All of the above.

* listed incorrectly as "Transient iterations per point".

Save*: Save a set of outputs

General Form

```
save [all | output ...] .save [all | output ...]
```

Save a set of outputs, discarding the rest. If a node has been mentioned in a **save** command, it appears in the working plot after a run has completed, or in the rawfile if spice is run in batch mode. If a node is traced or plotted (see below) it is also saved. For backward compatibility, if there are **no** save commands given, all outputs are saved.

When the keyword "all" appears in the save command, all default values (node voltages and voltage source currents) are saved in addition to any other values listed.

Sens*: Run a sensitivity analysis

General Form

```
sens output_variable sens output_variable ac ( DEC | OCT | LIN ) N Fstart
Fstop
```

Perform a Sensitivity analysis. *output_variable* is either a node voltage (ex. "v(1)" or "v(A,out)") or a current through a voltage source (ex. "i(vtest)"). The first form calculates DC sensitivities, the second form calculates AC sensitivies. The output values are in dimensions of change in output per unit change of input (as opposed to percent change in output or per percent change of input).

Set: Set the value of a variable

General Form

```
set [word] set [word = value] ...
```

Set the value of **word** to be **value**, if it is present. You can set any word to be any value, numeric or string. If no value is given then the value is the boolean 'true'.

The value of *word* may be inserted into a command by writing *$word*. If a variable is set to a list of values that are enclosed in parentheses (which **must** be separated from their values by white space), the value of the variable is the list.

The variables used by nutmeg are listed in the following section.

Setcirc*: Change the current circuit

General Form

```
setcirc [circuit name]
```

The current circuit is the one that is used for the simulation commands below. When a circuit is loaded with the **source** command (see below) it becomes the current circuit.

Setplot: Switch the current set of vectors

General Form

```
setplot [plotname]
```

Set the **current plot** to the plot with the given name, or if no name is given, prompt the user with a menu. (Note that the plots are named as they are loaded, with names like **tran1** or **op2**. These names are shown by the **setplot** and **display** commands and are used by **diff**, below.) If the "New plot" item is selected, the current plot becomes one with no vectors defined.

Note that here the word "plot" refers to a group of vectors that are the result of one SPICE run. When more than one file is loaded in, or more than one plot is present in one file, **nutmeg** keeps them separate and only shows you the vectors in the current plot.

Settype: Set the type of a vector

General Form

```
settype type vector ...
```

Change the type of the named vectors to **type**. Type names can be found in the manual page for **sconvert**.

Shell: Call the command interpreter

General Form

```
shell [ command ]
```

Call the operating system's command interpreter; execute the specified command or call for interactive use.

Shift: Alter a list variable

General Form

```
shift [varname] [number]
```

If *varname* is the name of a list variable, it is shifted to the left by *number* elements (i.e, the *number* leftmost elements are removed). The default *varname* is **argv**, and the default *number* is 1.

Show*: List device state

General Form

```
show devices [ : parameters ] , ...
```

Old Form

```
show -v @device [ [ name ] ]
```

The **show** command prints out tables summarizing the operating condition of selected devices (much like the **spice2** operation point summary). If *device* is missing, a default set of devices are listed, if *device* is a single letter, devices of that type are listed; if *device* is a subcircuit name (beginning and ending in ":") only devices in that subcircuit are shown (end the name in a double-":" to get devices within sub-subcircuits recursively). The second and third forms may be combined ("letter:subcircuit:") or "letter:subcircuit::") to select a specific type of device from a subcircuit. A device's full name may be specified to list only that device. Finally, devices may be selected by model by using the form "#modelname" or ":subcircuit#modelname" or "letter:subcircuit#modelname".

If no *parameters* are specified, the values for a standard set of parameters are listed. If the list of *parameters* contains a "+", the default set of parameters is listed along with any other specified parameters.

For both *devices* and *parameters*, the word "all" has the obvious meaning. Note: there must be spaces separating the ":" that divides the *device* list from the *parameter* list.

The "old form" (with "-v") prints the data in a older, more verbose pre-spice3f format.

Showmod*: List model parameter values

General Form

```
showmod models [ : parameters ] , ...
```

The **showmod** command operates like the **show** command (above) but prints out model parameter values. The applicable forms for *models* are a single letter specifying the device type letter, "letter:subckt:", "modelname", ":subckt:modelname", or "letter:subcircuit:modelname".

Source: Read a Spice3 input file

General Form

```
source file
```

For Spice3: Read the Spice3 input file **file**. **Nutmeg** and Spice3 commands may be included in the file, and must be enclosed between the lines *.control* and *.endc*. These commands are executed immediately after the circuit is loaded, so a control line of *ac* ... works the same as the corresponding *.ac* card. The first line in any input file is considered a title line and not parsed but kept as the name of the circuit. The exception to this rule is the file *.spiceinit*. Thus, a Spice3 command script must begin with a blank line and then with a *.control* line. Also, any line beginning with the characters *# is considered a control line. This makes it possible to imbed commands in Spice3 input files that are ignored by earlier versions of Spice2

For Nutmeg: Reads commands from the file **filename.** Lines beginning with the character * are considered comments and ignored.

Status*: Display breakpoint information

General Form

```
status
```

Display all of the traces and breakpoints currently in effect.

Step*: Run a fixed number of timepoints

General Form

```
step [number]
```

Iterate **number** times, or once, and then stop.

Stop*: Set a breakpoint

General Form

```
stop [ after n] [ when value cond value ] ...
```

Set a breakpoint. The argument **after n** means stop after **n** iteration number **n**, and the argument **when** *value cond value* means stop when the first *value* is in the given relation with the second *value*, the possible relations being

```
eq or \\&= equal to
ne or <> not equal to
gt or > greater than
lt or < less than
ge or >= greater than or equal to
le or <= less than or equal to
```

IO redirection is disabled for the **stop** command, since the relational operations conflict with it (it doesn't produce any output anyway). The *value*s above may be node names in the running circuit, or real values. If more than one condition is given, e.g. **stop after 4 when v(1) > 4 when v(2) < 2**, the conjunction of the conditions is implied.

Tf*: Run a Transfer Function analysis

General Form

```
tf output_node input_source
```

The **tf** command performs a transfer function analysis, returning the transfer function (output/input), output resistance, and input resistance between the given output node and the given input source. The analysis assumes a small-signal DC (slowly varying) input.

Trace*: Trace nodes

General Form

```
trace [ node ...]
```

For every step of an analysis, the value of the node is printed. Several traces may be active at once. Tracing is not applicable for all analyses. To remove a trace, use the **delete** command.

Tran*: Perform a transient analysis

General Form

```
tran Tstep Tstop [ Tstart [ Tmax ] ] [ UIC ]
```

Perform a transient analysis. See the previous sections of this manual for more details.

Transpose: Swap the elements in a multi-dimensional data set

General Form

```
transpose vector vector ...
```

This command transposes a multidimensional vector. No analysis in Spice3 produces multidimensional vectors, although the DC transfer curve may be run with two varying sources. You must use the "reshape" command to reform the one-dimensional vectors into two dimensional vectors. In addition, the default scale is incorrect for plotting. You must plot versus the vector corresponding to the second source, but you must also refer only to the first segment of this second source vector. For example (circuit to produce the tranfer characteristic of a MOS transistor):

```
spice3 > dc vgg 0 5 1 vdd 0 5 1

spice3 > plot i(vdd)

spice3 > reshape all [6,6]
```

```
spice3 > transpose i(vdd) v(drain)

spice3 > plot i(vdd) vs v(drain)[0]
```

Unalias: Retract an alias

General Form

```
unalias [word ...]
```

Removes any aliases present for the **word**s.

Undefine: Retract a definition

General Form

```
undefine function
```

Definitions for the named user-defined functions are deleted.

Unset: Clear a variable

General Form

```
unset [word ...]
```

Clear the value of the specified variable(s) (*word*).

Version: Print the version of Spice

General Form

```
version [version id]
```

Print out the version of **nutmeg** that is running. If there are arguments, it checks to make sure that the arguments match the current version of SPICE. (This is mainly used as a **Command:** line in rawfiles.)

Where: Identify troublesome node or device

General Form

```
where
```

When performing a transient or operating point analysis, the name of the last node or device to cause non-convergence is saved. The **where** command prints out this information so that you can examine the circuit and either correct the problem or make a bug report. You may do this either in the middle of a run or after the simulator has given up on the analysis. For transient simulation, the **iplot** command can be used to monitor the progress of the analysis. When the analysis slows down severly or hangs, interrupt the simulator (with control-C) and issue the **where** command. Note that only one node or device is printed; there may be problems with more than one node.

Write: Write data to a file

General Form

```
write [file] [exprs]
```

Writes out the expressions to **file.**

First vectors are grouped together by plots, and written out as such (i.e, if the expression list contained three vectors from one plot and two from another, then two plots are written, one with three vectors and one with two). Additionally, if the scale for a vector isn't present, it is automatically written out as well.

The default format is ascii, but this can be changed with the **set filetype** command. The default filename is **rawspice.raw**, or the argument to the **-r** flag on the command line, if there was one, and the default expression list is **all**.

Xgraph: use the xgraph(1) program for plotting.

General Form

```
xgraph file [exprs] [plot options]
```

The spice3/nutmeg xgraph command plots data like the **plot** command but via **xgraph**, a popular X11 plotting program.

If *file* is either "temp" or "tmp" a temporary file is used to hold the data while being plotted. For available plot options, see the **plot** command. All options except for polar or smith plots are supported.

CONTROL STRUCTURES

While End

General Form

```
while condition statement ... end
```

While *condition*, an arbitrary algebraic expression, is true, execute the statements.

Repeat End

General Form

```
repeat [number] statement ... end
```

Execute the statements *number* times, or forever if no argument is given.

Dowhile End

General Form

```
dowhile condition statement ... end
```

The same as **while**, except that the *condition* is tested after the statements are executed.

Foreach End

General Form

```
foreach var value ... statement ... end
```

The statements are executed once for each of the *value*s, each time with the variable *var* set to the current one. (*var* can be accessed by the $*var* notation see below).

If Then Else

General Form

```
if condition statement ... else statement ... end
```

If the *condition* is non-zero then the first set of statements are executed, otherwise the second set. The **else** and the second set of statements may be omitted.

Label

General Form

```
label word
```

If a statement of the form goto *word is encountered, control is transferred to this point, otherwise this is a no-op.*

Goto

General Form

```
goto word
```

If a statement of the form label *word is present in the block or an enclosing block, control is transferred there. Note that if the label is at the top level, it must be before the goto statement (i.e, a forward goto may occur only within a block).*

Continue

General Form

```
continue
```

If there is a **while, dowhile,** or **foreach** block enclosing this statement, control passes to the test, or in the case of **foreach**, the next value is taken. Otherwise an error results.

Break

General Form

```
break
```

If there is a **while, dowhile,** or **foreach** block enclosing this statement, control passes out of the block. Otherwise an error results.

Of course, control structures may be nested. When a block is entered and the input is the terminal, the prompt becomes a number of >'s corresponding to the number of blocks the user has entered. The current control structures may be examined with the debugging command **cdump.**

VARIABLES

The operation of both **Nutmeg** and **Spice3** may be affected by setting variables with the "set" command. In addition to the variables mentioned below, the **set** command in **Spice3** also affect the behaviour of the simulator via the options previously described under the section on ".OPTIONS".

The variables meaningful to **nutmeg** which may be altered by the **set** command are:

diff_abstol	he absolute tolerance used by the **diff** command.
appendwrite	Append to the file when a **write** command is issued, if one already exists.
colorN	These variables determine the colors used, if **X** is being run on a color display.IN may be between 0 and 15. Color 0 is the background, color 1 is the grid and text color, and colors 2 through 15 are used in order for vectors plotted. The value of the **color** variables should be names of colors, which may be found in the file **/usr/lib/rgb.txt**.
combplot	Plot vectors by drawing a vertical line from each point to the X-axis, as opposed to joining the points.Note that this option is subsumed in the plottype option, below.
cpdebug	Print cshpar debugging information (must be complied with the -DCPDEBUG flag). Unsupported in the current release
debug	If set then a lot of debugging information is printed (must be compiled with the -DFTEDEBUG flag).Unsupported in the current release.
device	The name (/dev/tty??) of the graphics device.If this variable isn't set then the user's terminal is used.To do plotting onanother monitor you probably have to set both the **device** and **term** variables.(If **device** is set to the name of a file, **nutmeg** dumps the graphics control codes into this file -- this is useful for saving plots.)
echo	Print out each command before it is executed.
filetype	This can be either **ascii** or **binary**, and determines what format rawfiles are. The default is **ascii**.
fourgridsize	How many points to use for interpolating into when doing fourier analysis.
gridsize	If this variable is set to an integer, this number is used as the number of equally spaced points to use for the Y-axis when plotting. Otherwise the current scale is used (which may not have equally spaced points). If the current scale isn't strictly monotonic, then this option has no effect.
hcopydev	If this is set, when the **hardcopy** command is run the resulting file is automatically printed on the printer named **hcopydev** with the command lpr -P**hcopydev** -g file.
hcopyfont	This variable specifies the font name for hardcopy output plots. The value is device dependent.
hcopyfontsize	This is a scaling factor for the font used in hardcopy plots.
hcopydevtype	This variable specifies the type of the printer output to use in the **hardcopy** command. If hcopydevtype is not set, plot (5) format is assumed. The standard distribution currently recognizes **postscript** as an alternative output format. When used in conjunction with **hcopydev**, h**copydevtype** should specify a format supported by the printer.
hcopyfontsize	This is a scaling factor for the font used in hardcopy plots.
hcopydevtype	This variable specifies the type of the printer output to use in the **hardcopy** command. If hcopydevtype is not set,

	plot (5) format is assumed. The standard distribution currently recognizes **postscript** as an alternative output format. When used in conjunction with **hcopydev**, h**copydevtype** should specify a format supported by the printer.
height	The length of the page for **asciiplot** and **print col**.
history	The number of events to save in the history list.
lprplot5	This is a printf(3s) style format string used to specify the command to use for sending plot(5)-style plots to a printer or plotter. The first parameter supplied is the printer name, the second parameter supplied is a file name containing the plot. Both parameters are strings. It is trivial to cause Spice3 to abort by supplying a unreasonable format string.
lprps	This is a printf(3s) style format string used to specify the command to use for sending PostScript plots to a printer or plotter. The first parameter supplied is the printer name, the second parameter supplied is a file name containing the plot. Both parameters are strings. It is trivial to cause Spice3 to abort by supplying a unreasonable format string.
nfreqs	The number of frequencies to compute in the **fourier** command. (Defaults to 10.)
nobreak	Don't have **asciiplot** and **print col** break between pages.
noasciiplotvalue	Don't print the first vector plotted to the left when doing an **asciiplot**.
noclobber	Don't overwrite existing files when doing IO redirection.
noglob	Don't expand the global characters `*', `?', `[', `and `]'. This is the default.
nogrid	Don't plot a grid when graphing curves (but do label the axes).
nomoremode	If **nomoremode** is not set, whenever a large amount of data is being printed to the screen (e.g, the **print** or **asciiplot** commands), the output is stopped every screenful and continues when a carriage return is typed. If **nomoremode** is set then data scrolls off the screen without check.
nonomatch	If **noglob** is unset and a global expression cannot be matched, use the global characters literally instead of complaining.
nosort	Don't have **display** sort the variable names.
noprintscale	Don't print the scale in the leftmost column when a print col command is given.
numdgt	The number of digits to print when printing tables of data (**fourier**, **print col**). The default precision is 6 digits. On the VAX, approximately 16 decimal digits are available using double precision, so **numdgt** should not be more than 16. If the number is negative, one fewer digit is printed to ensure constant widths in tables.
plottype	This should be one of **normal**, **comb**, or **point**:chars. **normal**, the default, causes points to be plotted as parts of connected lines. **comb** causes a comb plot to be done (see the description of the **combplot** variable above). **point** causes each point to be plotted separately - the **chars** are a list

	of characters that are used for each vector plotted. If they are omitted then a default set is used.
polydegree	The degree of the polynomial that the **plot** command should fit to the data. If polydegree is N, then **nutmeg** fits a degree N polynomial to every set of N points and draw 10 intermediate points in between each endpoint. If the points aren't monotonic, then it tries rotating the curve and reducing the degree until a fit is achieved.
polysteps	The number of points to interpolate between every pair of points available when doing curve fitting. The default is 10.
program	The name of the current program (argv[0])
prompt	The prompt, with the character `!' replaced by the current event number.
rawfile	The default name for rawfiles created.
diff_reltol	The relative tolerance used by the **diff** command.
remote_shell	Overrides the name used for generating rspice runs (default is "rsh").
rhost	The machine to use for remote SPICE-3 runs, instead of the default one (see the description of the **rspice** command, below).
rprogram	The name of the remote program to use in the **rspice** command.
slowplot	Stop between each graph plotted and wait for the user to type return before continuing.
sourcepath	A list of the directories to search when a **source** command is given. The default is the current directory and the standard spice library (/usr/local/lib/spice, or whatever **LIBPATH** is #defined to in the Spice3 source.
spicepath	The program to use for the **aspice** command. The default is /cad/bin/spice.
term	The mfb name of the current terminal.
units	If this is **degrees**, then all the trig functions will use degrees instead of radians.
unixcom	If a command isn't defined, try to execute it as a UNIX command. Setting this option has the effect of giving a **rehash** command, below. This is useful for people who want to use **nutmeg** as a login shell.
verbose	Be verbose. This is midway between **echo** and **debug** / **cpdebug**.
diff_vntol	The absolute voltage tolerance used by the **diff** command.

width	The width of the page for **asciiplot** and **print col**.
x11linearcs	Some X11 implementations have poor arc drawing. If you set this option, Spice3 will plot using an approximation to the curve using straight lines.
xbrushheight	The height of the brush to use if **X** is being run.
xbrushwidth	The width of the brush to use if **X** is being run.
xfont	

	The name of the X font to use when plotting data and entering labels. The plot may not look good if this is a variable-width font.

There are several **set** variables that Spice3 uses but Nutmeg does not. They are:

editor	The editor to use for the **edit** command.
modelcard	The name of the model card (normally **.model**).
noaskquit	Do not check to make sure that there are no circuits suspended and no plots unsaved. Normally Spice-3 warns the user when he tries to quit if this is the case.
nobjthack	Assume that BJTs have 4 nodes.
noparse	Don't attempt to parse input files when they are read in (useful for debugging). Of course, they cannot be run if they are not parsed.
nosubckt	Don't expand subcircuits
renumber	Renumber input lines when an input file has **.include**'s.
subend	The card to end subcircuits (normally **.ends**).
subinvoke	The prefix to invoke subcircuits (normally **x**).
substart	The card to begin subcircuits (normally **.subckt**).

MISCELLANEOUS

If there are subcircuits in the input file, Spice3 expands instances of them. A subcircuit is delimited by the cards **.subckt** and **.ends,** or whatever the value of the variables **substart** and **subend** is, respectively. An instance of a subcircuit is created by specifying a device with type 'x' ' the device line is written

```
xname node1 node2 ... subcktname
```

where the nodes are the node names that replace the formal parameters on the .**subckt** line. All nodes that are not formal parameters are prepended with the name given to the instance and a ':', as are the names of the devices in the subcircuit. If there are several nested subcircuits, node and device names look like **subckt1:subckt2:...:name**. If the variable **subinvoke** is set, then it is used as the prefix that specifies instances of subcircuits, instead of 'x'.

Nutmeg occasionally checks to see if it is getting close to running out of space, and warns the user if this is the case. (This is more likely to be useful with the SPICE front end.)

C-shell type quoting with "" and '', and backquote substitution may be used. Within single quotes, no further substitution (like history substitution) is done, and within double quotes, the words are kept together but further substitution is done. Any text between backquotes is replaced by the result of executing the text as a command to the shell.

Tenex-style ('set filec' in the 4.3 C-shell) command, filename, and keyword completion is possible: If EOF (control-D) is typed after the first character on the line, a list of the commands or possible arguments is printed (If it is alone on the line it exits **nutmeg**). If escape is typed, then

nutmeg trys to complete what the user has already typed. To get a list of all commands, the user should type &ltspace;> ^D.

The values of variables may be used in commands by writing **$varname** where the value of the variable is to appear. The special variables *$$* and *$<* refer to the process ID of the program and a line of input which is read from the terminal when the variable is evaluated, respectively. If a variable has a name of the form **$&word;**, then **word** is considered a vector (see above), and its value is taken to be the value of the variable. If *$foo* is a valid variable, and is of type **list**, then the expression *$foo[low-high]* represents a range of elements. Either the upper index or the lower may be left out, and the reverse of a list may be obtained with *$foo[len-0]*. Also, the notation *$?foo* evaluates to 1 if the variable *foo* is defined, 0 otherwise, and *$#foo* evaluates to the number of elements in *foo* if it is a list, 1 if it is a number or string, and 0 if it is a boolean variable.

History substitutions, similar to C-shell history substitutions, are also available see the C-shell manual page for all of the details.

The characters ~, {, and } have the same effects as they do in the C-Shell, i.e., home directory and alternative expansion. It is possible to use the wildcard characters *, ?, [, and] also, but only if you **unset noglob** first. This makes them rather useless for typing algebraic expressions, so you should **set noglob** again after you are done with wildcard expansion. Note that the pattern **[^abc]** matchs all characters *except* **a, b,** *and* **c.**

IO redirection is available the symbols **>, >>, >&, >>&,** and**<** have the same effects as in the C-shell.

You may type multiple commands on one line, separated by semicolons.

If you want to use a different **mfbcap** file than the default (usually **~cad/lib/mfbcap**), you have to set the environment variable **SPICE_MFBCAP** before you start **nutmeg** or **spice**. The **-m** option and the **mfbcap** variable no longer work.

If **X** is being used, the cursor may be positioned at any point on the screen when the window is up and characters typed at the keyboard are added to the window at that point. The window may then be sent to a printer using the **xpr(1)** program.

Nutmeg can be run under VAX/VMS, as well as several other operating systems. Some features like command completion, expansion of *, ?, and [], backquote substitution, the shell command, and so forth do not work.

On some systems you have to respond to the *-more-* prompt during plot with a carriage return instead of any key as you can do on UNIX.

BUGS

The label entry facilities are primitive. You must be careful to type slowly when entering labels -- **nutmeg** checks for input once every second, and can get confused if characters arrive faster.

If you redefine colors after creating a plot window with X, and then cause the window to be redrawn, it does not redraw in the correct colors.

When defining aliases like

```
alias pdb plot db( '!:1' - '!:2' )
```

you must be careful to quote the argument list substitutions in this manner. If you quote the whole argument it might not work properly.

In a user-defined function, the arguments cannot be part of a name that uses the *plot.vec* syntax. For example:

```
define check(v(1)) cos(tran1.v(1))
```

does **not** work.

If you type **plot all all**, or otherwise use a wildcard reference for one plot twice in a command, the effect is unpredictable.

The **asciiplot** command doesn't deal with log scales or the **delta** keywords.

Often the names of terminals recognized by **MFB** are different from those in /etc/termcap. Thus you may have to reset your terminal type with the command

```
set term = termname
```

where **termname** is the name in the **mfbcap** file.

The **hardcopy** command is useless on VMS and other systems without the **plot** command, unless the user has a program that understands *plot(5)* format.

Spice3 recognizes all the notations used in SPICE2 **.plot** cards, and translates **vp(1)** into **ph(v(1))**, and so forth. However, if there are spaces in these names it won't work. Hence **v(1, 2)** and **(-.5, .5)** aren't recognized.

BJTs can have either 3 or 4 nodes, which makes it difficult for the subcircuit expansion routines to decide what to rename. If the fourth parameter has been declared as a model name, then it is assumed that there are 3 nodes, otherwise it is considered a node. To disable this, you can set the variable "nobjthack" which forces BJTs to have 4 nodes (for the purposes of subcircuit expansion, at least).

The **@name[param]** notation might not work with **trace, iplot,** etc. yet.

- Parent Directory -