A Short SPICE Tutorial

Kenneth H. Carpenter
Department of Electrical and Computer Engineering
Kanas State University

September 15, 2003 - November 10, 2004

1 Introduction

SPICE is an acronym for "Simulation Program with Integrated Circuit Emphasis." There are many versions of this program, but all are based on the work done at the University of California at Berkeley[1], which resulted in two major, final versions: SPICE2G6 and SPICE3F4. Others have enhanced and extended (and fixed bugs) in these to produce new versions. Some of these are commercial; some are free software. In the following tutorial, the "lowest common denominator" of all the SPICE versions is described. That is, the SPICE input files will work with all the versions (known to the author). (Comments on how to extend the input files to take advantage of special features of special versions will be given in a few places.)

SPICE uses a modified nodal analysis method[2] to solve electrical circuits. To communicate the topology of the circuit and the circuit values to SPICE one requires an input file containing a *netlist*. To communicate the analyses one requires from the program one places control lines in the input file. To obtain output from the analyses one places output control lines in the input file. The details of constructing this input file follow.

2 SPICE input file syntax

All SPICE circuit simulators require an input file. This input file is an ASCII text file. The input file is line oriented.

2.1 Lines in a SPICE input file

- The first line of the file is a title to be placed on the output. It can contain any text.
  DO NOT FORGET that the first line in the file will be taken as the title and will not be used by SPICE in any other way, even if it is otherwise a valid control or netlist line.

- Any line that has an asterisk (*) in the first column is a comment. The entire line is ignored by SPICE (except for sometimes printing it in an output listing).

- Lines that are part of the netlist begin with an alphabetic character; the remainder of the line depends on the circuit element represented.

- Lines that control the SPICE program have a period (.) in the first column.

- Lines that have a plus sign (+) in the first column continue the previous line. That is, the SPICE program combines the characters in the line beginning with the plus sign with the previous line by discarding both the plus sign and the new line character(s) of the previous line. (It is wise to make continuation lines start where there is a natural break in the text.)

Table 1 gives the SPICE input file for simulation of the circuit in Fig. 1. Table 1 illustrates the syntax described here.

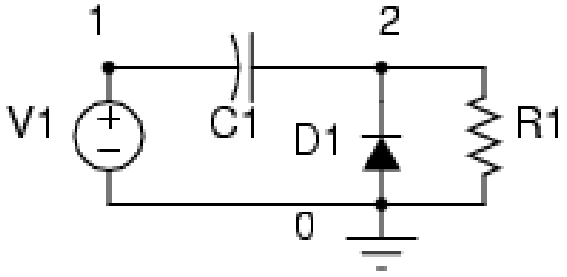

Fig. 1: Circuit schematic diagram for simulation of Table 1.

```
SPICE sample circuit - diode clamp

*comment lines begin with asterisks

*independent voltage source with DC value, AC value, and
*transient square wave value:
V1 1 0 1 AC 1 pulse -10 20 0 1.e-8 1.e-8 1e-3 2e-3
*the square wave defined above has -10V to +20V extent, with
*a period of 2 milliseconds

*capacitor for clamping
C1 1 2 1e-6

*diode for clamp - model name is dclamp
D1 2 0 dclamp

*load resistor - large enought that RC >> 2 ms
R1 2 0 1e5

*model for diode
.model dclamp D(IS=1e-14)

*DC transfer function generated for this circuit
.DC V1 -20 20 .1

*AC frequency sweep - assumes circuit is biased with V1 = 1V
.AC OCT 20 10 1e2 1e4
*frequency is swept logarithmically from 100Hz to 10000Hz

*transient analysis will show clamping
.TRAN 1e-4 8e-3 0 1e-5
*start at time zero, go for 8 ms, make internal steps 10 microsec
*save print data at .1 ms intervals

*that's all folks
.end
```

Table 1: SPICE input file for sample simulation.

2.2 Divisions of a line in the SPICE input file

The parser separates the characters on the input line into tokens. Tokens may be separated by

- blanks (space characters)
- an equals sign (=) (but only use these when you are giving a value to a parameter)
- a left or right parenthesis (but only use these when you are grouping following tokens)
- extra spaces are ignored

Tokens can be either

- strings – sequences of non-separator characters
  or
- numbers – floating or fixed point decimal values as in "c" or FORTRAN
  e.g., 1.23e-5 or 0.0000123
  (Powers of ten can be indicated by adding one or more letters to the end of the number, without any intervening spaces, but don't do this as a beginner! See the *SPICE User's Manual*[3] for details.)

3 The SPICE netlist

The part of the SPICE input file called the *netlist* describes the circuit to spice. In order to write a netlist one must have a circuit schematic with all nodes labeled.

- All nodes must have labels – even nodes that connect only two circuit elements.
- There must be a ground node, and it must be labeled zero.
- For compatibility with all versions of SPICE, the node labels must be positive integers (except for ground which is node 0).

See Fig. 1 for an example of placing node labels on a schematic.

Each circuit element must have a text label. The first letter of the label must be in the first column of the line for the circuit element in the netlist. The first letter specifies the type of circuit element it is. The rest of the letters and/or numbers of the label distinguish the different circuit elements of the same kind. Always use at least two characters for a circuit element label. These labels should be placed on the circuit schematic as in Fig. 1.

The types of circuit elements and the first letters of their labels are as follows:

V independent voltage source

I independent current source

R resistor

C capacitor

L inductor

D diode

Q bipolar junction transistor

J JFET

M MOSFET

T lossless transmission line

There are also letters to designate subcircuits, dependent sources, transmission lines, etc. See the *SPICE User's Manual*[3] or the on-line help files of the various SPICE versions for details of these other elements.

A line in a SPICE input file representing a circuit element in the netlist has the following parts:

- The element label is the string identifying the element. It must begin with the letter identifying the element type in columm one.
- After the circuit element label comes one or more spaces, then the label of the first node connected to the circuit element.

- Then comes one or more spaces, then the label of the second node connected to the circuit element.
- If the circuit element connects to more than two nodes, then there are space separated node labels for all the nodes.
- After the node labels comes one or more spaces followed by the circuit element's value. Just how this value is expressed depends on the particular element. See below, or the *SPICE User's Manual*[3] or the on-line help files of the various SPICE versions for details of the values.

3.1 Value fields for circuit element lines

3.1.1 R, L, and C lines

These devices have as the first token the value field the number of their value. (in SI units - Ohms, Henries, or Farads).

L and C elements may also have their initial conditions specified for transient analysis by a parameter token, IC, followed by a number token giving the initial value.

3.1.2 T lines

A lossless transmission line requires specification of four nodes: two at the beginning of the line and two at the end. It also requires specification of the characteristic impedance and time delay of the line by giving them as parameters in the value field. A typical line in the netlist would be

```
T1 1 0 2 0 Z0=50 TD=1E-6
```

This line has nodes 1 and 0 at the beginning and 2 and 0 at the end. The characteristic impedance is 50Ω and the time delay is 1μs.

3.2 V and I lines

Independent sources can have a complicated set of value parameters given.

The DC value of a source is either the first number after the last node label on the line, or it is the number following the token "DC".

The peak magnitude of the AC value of a source is the number following the token "AC". (If the next token after "AC" is not a number then the

magnitude is taken as 1 and the phase as 0.) If the next token after the magnitude number is a number, it is the phase of the phasor. If the next token is not a number, the phase is zero. *If a source is to be present in a circuit but have an AC value of zero, then the "AC" token must not appear on the line.*

A source to be used in a transient analysis has a function of time specified as the last set of tokens on the line. This function of time has several possible forms. See the *SPICE User's Manual*[3] for details.

A single source can have any or all of the DC, AC, and function of time values.

3.3 D, Q, J and M lines

These semiconductor devices must have models. The first token after the last node label must be a string which identifies the model. Following this token there can be other strings and numbers which modify the parameters of the model and set initial conditions. See the *SPICE User's Manual*[3] for details.

When a line for each circuit element in the schematic is present then the netlist is complete. The Table 1 for an example.

4 Control lines in the SPICE input file

All other lines, besides comments, title, and netlist lines, in the SPICE input file are *control lines*. Control lines all have a period (.) in the first column. There are many different control lines. They can be grouped as follows:

- .end line – must be the last line in the input file
- analysis lines – specify what simulations are to be performed
- include lines – used to nest input files (not needed in simple applications)
- option lines – set parameters and conditions for simulations, models, etc.
- output lines – determine how output from simulations is to be presented

- .model lines – define the strings used as model parameters for circuit elements in the netlist

4.1 .end line

Although some SPICE versions may work without it, most require that a line consisting of just ".end" be the last line in the input file.

4.2 analysis lines

There are many possible types of analyses. See the *SPICE User's Manual*[3] for all of them and for details of the following ones.

- .DC line – this causes a sweep of one or more voltage source DC values to be performed and the corresponding node voltages and source currents to be calculated to yield DC transfer functions.

  ```
  .DC V1 -5 12 .1
  ```

  would cause the source labeled "V1" to have its value swept from -5 volts to +12 volts in steps of .1 volt. The DC value on the V1 line of the netlist would be ignored.

- .AC line – this causes a frequency sweep of all the independent sources having an "AC" token.

  ```
  .AC V1 LIN 5 30 130
  ```

  would cause phasor analysis of the circuit to be performed for frequencies from 30 to 130 Hertz. 5 frequencies would be used. They would be separated linearly, that is by equal increments. Values of node voltages and source currents will be saved as complex numbers.

- .TRAN line – this causes a transient analysis to be performed.

  ```
  .TRAN .0001 .01 .05 .00001
  ```

  would cause time to be varied from an initial value of 0.05 seconds in internal steps not to exceed 0.00001 seconds until the stop time of 0.01 seconds is reached. Printed output of node voltages and currents can be obtained in steps of 0.001 seconds.
  If a token "UIC" is appended, then the initial conditions specified in the netlist will be used. Otherwise a DC analysis will be performed to obtain the initial conditions.

4.3 .model lines

When a model is specified on a netlist line, there must be a corresponding `.model` line in the input file to define that model. The tokens on that line are as follows:

- The first token on the line is `.model`
- The second token on the line is the same string as used in the circuit element line of the netlist to name the model.
- The third token on the line identifies what kind of a model it is.

D	for diode
NPN	for NPN BJT
PNP	for PNP BJT
NMOS	for n channel MOSFET

and so on.

The remaining tokens on the line give the model parameters values when other than the defaults are required. See the *SPICE User's Manual*[3] for the parameters that go with each model and for their default values.

4.4 output lines

4.4.1 .print

To obtain printed sets of node voltages and currents from simulations one includes control cards beginning with `.print`. For example

`.print AC V(2)` will cause a table of values to be placed in the SPICE output file having the frequencies used in the AC analysis in the first column and the complex values of the voltage at node 2 in the second column.

4.4.2 .plot

Lines beginning with `.plot` work like `.print` lines except that instead of a table of values a "printer plot" is placed in the output file.

This method of obtaining plots of output values is obsolete. Today one will use a graphical post processing program to produce the plots. With

PSpice, one puts a control line with the single token `.probe` into the input file in order to generate the output needed by the post processor, which is called Probe. BUT A LINE `.probe` MAY NOT APPEAR in the input file for any other verions of SPICE. Other versions of SPICE automatically output a file for graphical post processing without a line in the input file.

4.5 option lines

There are many options that may be given to SPICE. These are too numerous to mention in a beginning tutorial. Most simulations will not need any of them. See the *SPICE User's Manual*[3] for a list of them.

4.6 include lines

These lines allow other files to be nested inside the SPICE input file. The syntax of these lines depends on the particular versio of SPICE and the operating system. They are not needed for simple simulations.

5 Sample problem

The SPICE input file of Table 1, based on the circuit shown in Fig. 1, can be input to a version of SPICE and the simulations in the control lines carried out. When this is done using PSpice, the output file shown in Table 5 is obtained. If a `.probe` line had been included in the input file, then the graphical data saved for the DC transfer function anaylsis could be plotted with Probe to obtain the plot in Fig. 2.

```
**** 09/15/03 10:01:23 ********** Evaluation PSpice (Nov 1999) **************

 SPICE sample circuit - diode clamp

 ****     CIRCUIT DESCRIPTION

******************************************************************************



*comment lines begin with asterisks
```

```
*independent voltage source with DC value, AC value, and
*transient square wave value:
V1 1 0 1 AC 1 pulse -10 20 0 1.e-8 1.e-8 1e-3 2e-3
*the square wave defined above has -10V to +20V extent, with
*a period of 2 milliseconds

*capacitor for clamping
C1 1 2 1e-6

*diode for clamp - model name is dclamp
D1 2 0 dclamp

*load resistor - large enought that RC >> 2 ms
R1 2 0 1e5

*model for diode
.model dclamp D(IS=1e-14)

*DC transfer function generated for this circuit
.DC V1 -20 20 .1

*AC frequency sweep - assumes circuit is biased with V1 = 1V
.AC OCT 20 10 1e2
*frequency is swept logarithmically from 100Hz to 10000Hz

*transient analysis will show clamping
.TRAN 1e-4 8e-3 0 1e-5
*start at time zero, go for 8 ms, make internal steps 10 microsec
*save print data at .1 ms intervals

.probe

*that's all folks
.end

**** 09/15/03 10:01:23 ********** Evaluation PSpice (Nov 1999) **************

 SPICE sample circuit - diode clamp


 ****     Diode MODEL PARAMETERS


******************************************************************************



               dclamp
          IS   10.000000E-15


**** 09/15/03 10:01:23 ********** Evaluation PSpice (Nov 1999) **************

 SPICE sample circuit - diode clamp
```

```
 ****     SMALL SIGNAL BIAS SOLUTION       TEMPERATURE =   27.000 DEG C

*******************************************************************************

 NODE   VOLTAGE     NODE   VOLTAGE     NODE   VOLTAGE     NODE   VOLTAGE

(    1)    1.0000  (    2)-90.12E-27


    VOLTAGE SOURCE CURRENTS
    NAME         CURRENT

    V1           0.000E+00

    TOTAL POWER DISSIPATION   0.00E+00  WATTS

**** 09/15/03 10:01:23 ********** Evaluation PSpice (Nov 1999) **************

 SPICE sample circuit - diode clamp

 ****     INITIAL TRANSIENT SOLUTION       TEMPERATURE =   27.000 DEG C

*******************************************************************************

 NODE   VOLTAGE     NODE   VOLTAGE     NODE   VOLTAGE     NODE   VOLTAGE

(    1)  -10.0000  (    2)-90.12E-27


    VOLTAGE SOURCE CURRENTS
    NAME         CURRENT

    V1           0.000E+00

    TOTAL POWER DISSIPATION   0.00E+00  WATTS


          JOB CONCLUDED

          TOTAL JOB TIME             .75
```

Table 2. PSpice output file when file of Table 1 (with a `.probe` line added) is the input file.

If the SPICE input file of Table 1 is run with WinSpice or Spice3 then there is no text output file generated. Instead to see the printed values obtained with PSpice by the `.print` statement, one gives the `print` command interactively. To see the plot like that from probe, one gives the `plot` command interactively. Table 3 contains the terminal session from using spice3f5 under Linux. It shows how the printed values may be observed and how plots are called. The plot of `v(2) vs v(1)` for the transient analysis is shown in Fig. 3.

The student should try the input file of Table 1 in all three versions of SPICE to compare them. Then read further in the *SPICE User's Manual* and consult the web pages linked to the class web page for more information.

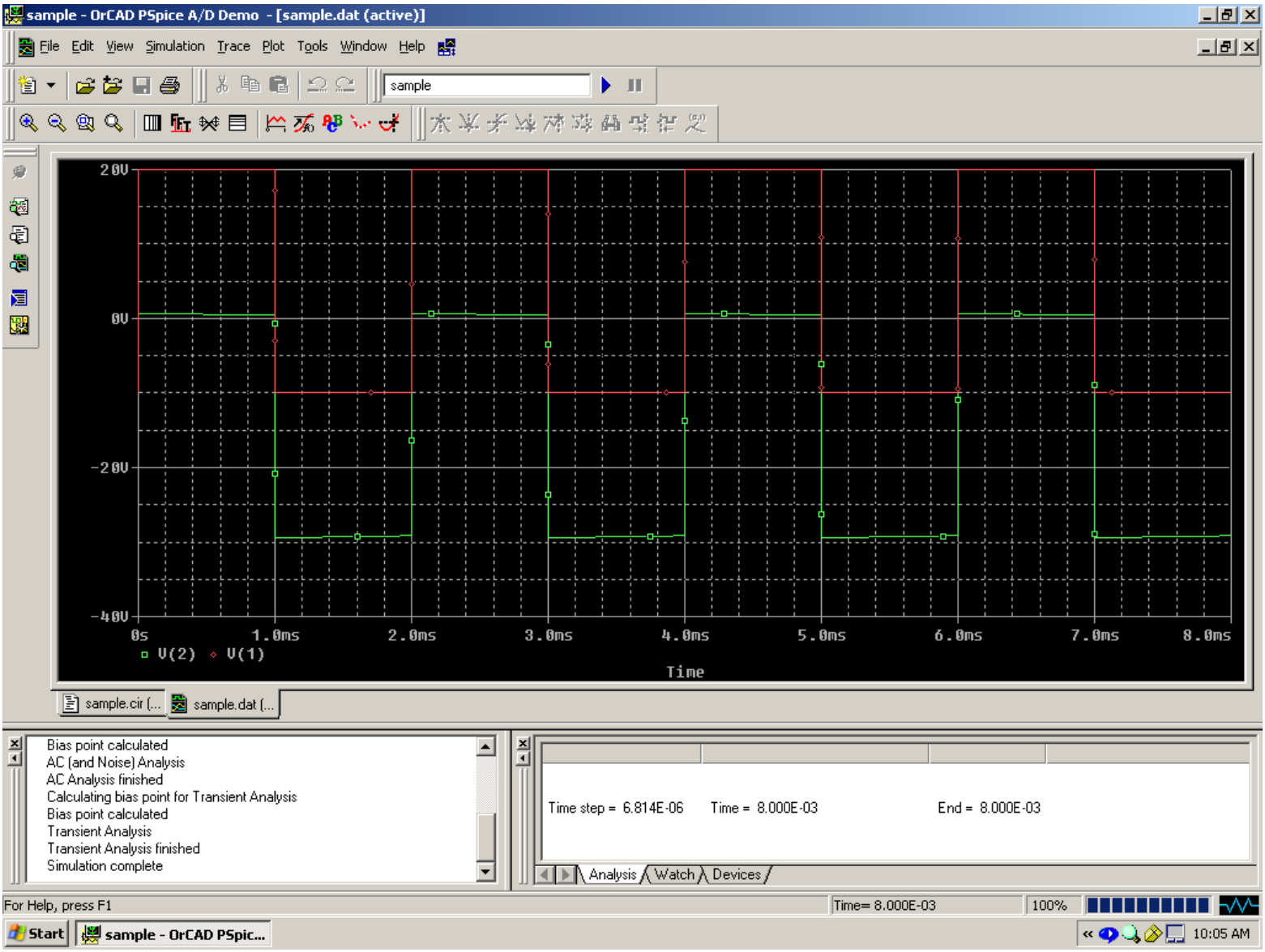

Fig. 2: Probe screen showing transient simulation from PSpice when input file is that given in Table 1 (with a `.probe` line added).

```
Script started on Mon Sep 15 11:12:23 2003
elmo{khc}~/classes/525/spicetut:1> spice3 sample.cir
Program: Spice, version: 3f5
Date built: Wed Aug 25 16:14:15 CDT 1999

Type "help" for more information, "quit" to leave.


Circuit: SPICE sample circuit - diode clamp

Spice 1 -> run
Spice 2 -> setplot
        Type the name of the desired plot:

        new     New plot
Current tran1   SPICE sample circuit - diode clamp (Transient Analysis)
        dc1     SPICE sample circuit - diode clamp (DC transfer characteristic)
        ac1     SPICE sample circuit - diode clamp (AC Analysis)
        const   Constant values (constants)
? ac1
Spice 3 -> print v(2)
                      SPICE sample circuit - diode clamp
                      AC Analysis  Mon Sep 15 11:12:31  2003
--------------------------------------------------------------------------------
Index   frequency                       v(2)
--------------------------------------------------------------------------------
0       1.000000e+01,   0.000000e+00    9.752955e-01,   1.552231e-01
1       1.035265e+01,   0.000000e+00    9.769117e-01,   1.501841e-01
2       1.071773e+01,   0.000000e+00    9.784245e-01,   1.452929e-01
3       1.109569e+01,   0.000000e+00    9.798402e-01,   1.405468e-01
4       1.148698e+01,   0.000000e+00    9.811648e-01,   1.359428e-01
5       1.189207e+01,   0.000000e+00    9.824039e-01,   1.314779e-01
6       1.231144e+01,   0.000000e+00    9.835629e-01,   1.271491e-01
7       1.274561e+01,   0.000000e+00    9.846468e-01,   1.229533e-01
8       1.319508e+01,   0.000000e+00    9.856602e-01,   1.188873e-01
9       1.366040e+01,   0.000000e+00    9.866076e-01,   1.149479e-01
10      1.414214e+01,   0.000000e+00    9.874932e-01,   1.111321e-01
11      1.464086e+01,   0.000000e+00    9.883210e-01,   1.074365e-01
12      1.515717e+01,   0.000000e+00    9.890946e-01,   1.038580e-01
13      1.569168e+01,   0.000000e+00    9.898175e-01,   1.003935e-01
14      1.624505e+01,   0.000000e+00    9.904929e-01,   9.703995e-02
        -- hit return for more, ? for help -- q

Spice 4 -> setplot tran1
Spice 5 -> plot v(2) v(1)
Spice 6 -> quit
```

Table 3: Interactive session for spice3 simulation of input file in Table 1.

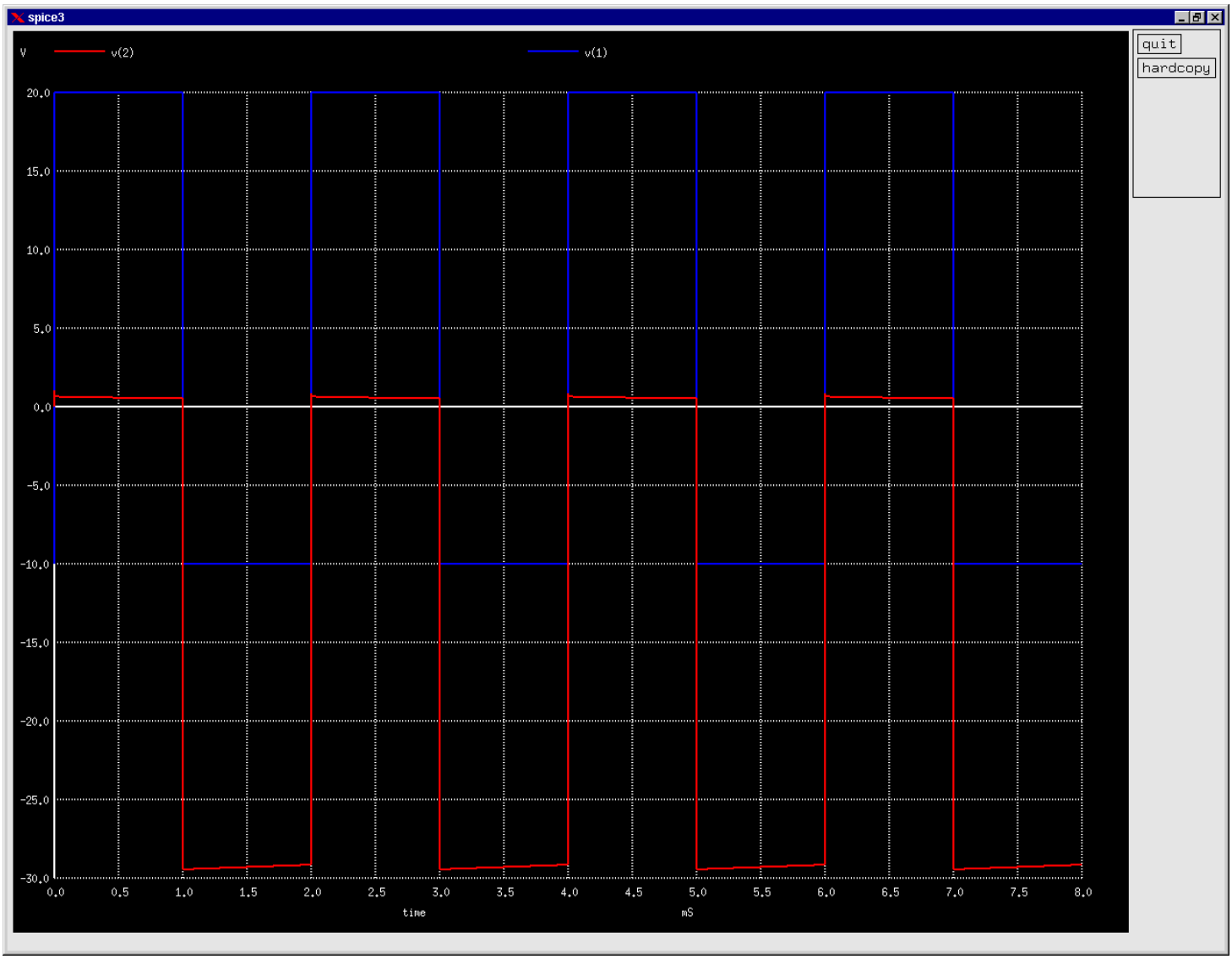

Fig. 3: Interactive plot from spice3 for transient simulation of input file in Table 1.

6 Transmission line sample problem

SPICE provides a convenient way to visualize transients on transmission lines. Suppose one wishes to examine voltage and currents as a function of time at the ends and midpoint of a line. Then the schematic of Fig. 4 may be used. In Fig. 4 the transmission line is simulated as two transmission

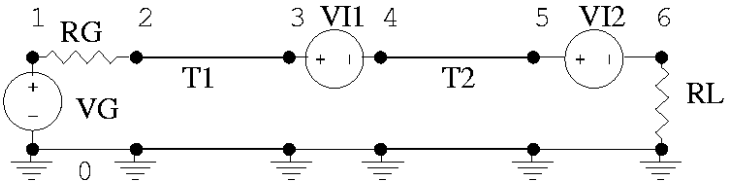

Fig. 4: Schematic for transmission line simulation using SPICE.

lines in series, each with half the length of the total line. The voltage source between the lines will act as an ammeter to show current halfway along the line. A netlist using this schematic with a 5 volt step function input for VG is given in Table 4. Running the input file in Table 4 with Spice3

```
Sample transmission line simulation with unit step input
VG 1 0 pwl 0 0 1e-10 5
RG 1 2 20
T1 2 0 3 0 z0=50 td=5e-9
VI1 3 4 0
T2 4 0 5 0 z0=50 td=5e-9
VI2 5 6 0
RL 6 0 100

.tran 1e-10 1e-7 0 1e-10
.end
```

Table 4: SPICE netlist for transmission line example, based on schematic of Fig. 4.

and plotting the current and voltage at the center of the line (by command `plot v(3) 100*i(vi1)`) yields the curves in Fig. 5.

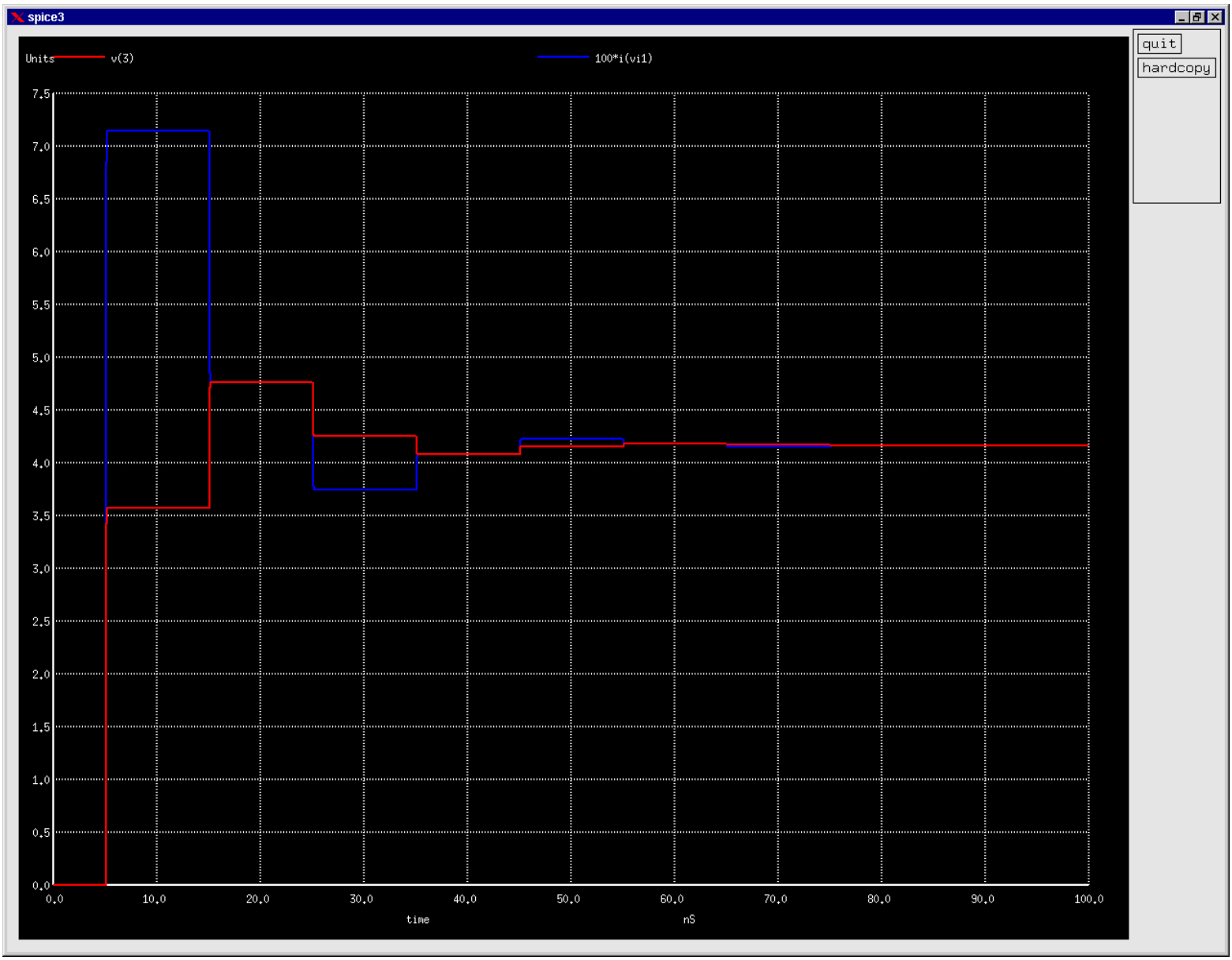

Fig. 5: Plot of voltage and current at midpoint of transmission line shown in Fig. 4 using Spice3 with the input file shown in Table 4.

References

[1] T. Perry, "Donald O. Pederson [electronic engineering biography]," *IEEE Spectrum*, Vol. 35, No. 6, pp. 22-27, June 1998.

[2] Laurence W. Nagel, "SPICE2: A Computer Program to Simulate Semiconductor Circuits," Memorandum No. ERL-M520, Electronics Research Laboratory, College of Engineering, University of California, Berkeley, May 9, 1975.

[3] T. Quarles, A. R. Newton, D.O.Pederson, A.Sangiovanni-Vincentelli, *SPICE3 Version 3f3 User's Manual*, Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, Ca.. May 1993.